REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (From - To)
)-11-2003	Final Technical Report	27 March, 2002 to 25 November, 2003

TITLE AND SUBTITLE
aval Battleforce Networks Composite Routing

5a. CONTRACT NUMBER N00014-02-C-0029
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

AUTHOR(S)
Jae H Kim (PI), Thomas R. Henderson, Phillip A. Spagnolo, Jeffrey M. Ahrenholz

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
ID ADDRESS(ES)
IANTOM WORKS
HE BOEING COMPANY
O. BOX 3707, MC 7L-49
EATTLE, WA 98124-2207

8. PERFORMING ORGANIZATION REPORT NUMBER

SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
FFICE OF NAVAL RESEARCH
SA FNC, CODE 311, BT1 RM 607-7
ALLSTON TOWER ONE, 800 NORTH
RLINGTON, VA 22217-5660
TTN: JOHN KUCHINSKI

10. SPONSOR/MONITOR'S ACRONYM(S)
ONR

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

. DISTRIBUTION / AVAILABILITY STATEMENT
ılimited

20040422 123

. SUPPLEMENTARY NOTES
iis ONR program is in support of Navy SPAWAR PMW-179 (ADNS)

. ABSTRACT

ııs document is the final technical report for the Boeing Composite Routing program as part of the Naval Battleforce Network (NBN) ogram, a subset of the ONR Knowledge Superiority and Assurance (KSA) Future Naval Capability (FNC) program

This document supplements the three previous technical reports and the software deliverables submitted under this contract:

- Naval Battleforce Networking Requirements Report;
- Protocol Evaluation Report,
- Final Demonstration Plan; and
- Routing Protocol Source Code and Documentation

particular, this document describes the software implementation of a modified OSPFv2 routing daemon for Linux, the laboratory testing sults of the prototype implementation, and the results of the final program demonstration in San Diego on September 18, 2003.

. SUBJECT TERMS
ireless OSPF (W-OSPF), Mobile Ad-hoc Network (MANET); Quality of Service (QoS); Differentiated Service (DiffServ)

. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Michael P. Steelsmith
REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (include area code)
Unclassified	Unclassified	Unclassified	Unclassified Unlimited	52	(206) 662-0738

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

BEST AVAILABLE COPY

FINAL TECHNICAL REPORT

NAVAL BATTLEFORCE NETWORK COMPOSITE ROUTING

OFFICE OF NAVAL RESEARCH
CONTRACT No. N00014-02-C-0029
CDRL ITEM – A007

SUBMITTED TO:

OFFICE OF NAVAL RESEARCH
KSA FNC, CODE 311, BT1 RM 607-7
BALLSTON TOWER ONE, 800 NORTH QUINCY STREET
ARLINGTON, VA 22217-5660

ATTN: JOHN KUCHINSKI
ONR PROGRAM MANAGER

NOVEMBER 21, 2003

DR. JAE H. KIM
BOEING PROGRAM MANAGER

PHANTOM WORKS
THE BOEING COMPANY
P.O.BOX 3707, MC/3W-51
SEATTLE, WA 98124-2207

This page is intentionally blank.

FINAL TECHNICAL REPORT

NAVAL BATTLEFORCE NETWORK COMPOSITE ROUTING

By

Thomas R. Henderson
Phillip A. Spagnolo
Jeffrey M. Ahrenholz

Phantom Works
The Boeing Company
P.O. Box 3707, MC/3W-51
Seattle, WA 98124-2207

November 21, 2003

FINAL TECHNICAL REPORT

NAVAL BATTLEFORCE NETWORK COMPOSITE ROUTING

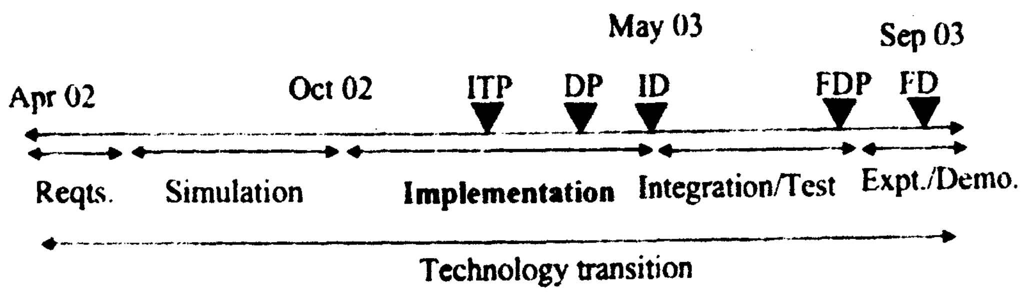
By

Thomas R. Henderson
Phillip A. Spagnolo
Jeffrey M. Ahrenholz

TABLE OF CONTENTS

1 Introduction

This document is the final technical report for the Boeing Composite Routing program as part of the Naval Battleforce Network (NBN) program, a subset of the ONR Knowledge Superiority and Assurance (KSA) Future Naval Capability (FNC) program. Figure 1-1 summarizes the program schedule and recent program deliverables for this contract.



Key:

ITP: Integration and Test Plan (not CDRL) **February 2003**

DP: Draft Demonstration Plan (not CDRL) **April 2003**

ID: Interim Demonstration (not CDRL) **15 May 2003**

FDP: Final Demonstration Plan (CDRL) **30 July 2003**

FD: Final Demonstration (CDRL) **18 September 2003**

Figure 1-1. Key recent program milestones for Boeing Composite Routing

This document supplements the three previous technical reports and the software deliverables submitted under this contract [1-4]:

- Naval Battleforce Networking Requirements Report;
- Protocol Evaluation Report;
- Final Demonstration Plan; and
- Routing Protocol Source Code and Documentation.

In particular, this document describes the software implementation of a modified OSPFv2 [5] routing daemon for Linux, the laboratory testing results of the prototype implementation, and the results of the final program demonstration in San Diego on September 18, 2003.

The remainder of this document is organized as follows:

- Section 2 lists applicable reference documents;
- Section 3 summarizes program deliverables;
- Section 4 provides an overview of the software implementation;
- Section 5 describes test and measurement techniques, including performance metrics and descriptions of hardware and software to be used for validation testing;
- Section 6 lists specific tests to which the Boeing implementation was subjected;
- Finally, Section 7 briefly mentions platform integration issues, which will be discussed more fully in the demonstration plan

2 Applicable documents

[1] "Naval Battleforce Networking Requirements Report," Boeing NBN CDRL-A002, submitted to Jerry Ferguson (SPAWAR/ONR), May 27, 2002.

[2] "Protocol Evaluation Report," Boeing NBN CDRL-A004, submitted to Jerry Ferguson (SPAWAR/ONR), October 31, 2002

[3] "Final Demonstration Plan," Boeing NBN CDRL-A005, submitted to Jerry Ferguson (SPAWAR/ONR), July 27, 2003.

[4] "Routing Protocol Source Code and Documentation," Boeing NBN CDRL-A006, submitted to Jerry Ferguson, September 30, 2003.

[5] "OSPF Version 2," Internet Engineering Task Force (IETF), Request for Comments: 2328, April 1998.

[6] Warner, C. et al., "Concept of Operations (CONOPS) and Architecture for the Battleforce Composite Network Project," SPAWAR Systems Center, August 30, 2002.

[7] Henderson, T. et al., "A Wireless Interface Type for OSPF," Proceedings of IEEE MILCOM 2003 Conference, Boston, MA, October 2003

3 Summary of program deliverables

Table 3-1 lists the data items that have been delivered under this contract, including those that were not officially a contractural deliverable (CDRL).

Date	Item	CDRL
27 May 2002	**Requirements Report**	**A002**
June 2002	Protocol Simulation Plan	No
June 2002	QualNet/OPNET Evaluation Report	No
31 October 2002	**Protocol Evaluation Report**	**A004**
February 2003	Draft Integration and Test Plan	No
April 2003	Draft Demonstration Plan	No
15 May 2003	Interim Demonstration	No
31 May 2003	Interim Demonstration Report	No
17 June 2003	Integration and Test Plan	No
17 June 2003	Internet-Draft specification of Wireless OSPF	No
17 June 2003	Internet-Draft specification of DSCP-based OSPF	No
17 June 2003	Internet-Draft specification of DSCP-based MOSPF	No
27 July 2003	**Final Demonstration Plan**	**A005**
18 Sept. 2003	Final Demonstration	No
26 Sept. 2003	**Routing Protocol Source Code and Documentation**	**A006**
26 Sept. 2003	(Draft) Final Technical Report	No
21 November 2003	**Final Technical Report**	**A007**
Various	**Presentation materials**	**A001**
Various	**Quarterly financial reports**	**A003**

Table 3-1 Summary of Boeing Composite Routing deliverables

4 Overview of implementation

4.1 General

The OSPFv2 Internet routing protocol is currently used in afloat Naval networks (the Automated Digital Network System, or ADNS). The OSPFv2 software developed for the NBN Composite Routing program contains the following extensions not found in commercial implementations, as agreed upon at the 7 November 2002 program review:

- **Development of a wireless interface type for OSPFv2.** As detailed in Reference 2, OSPFv2 performs poorly in a broadcast, wireless, single- or multi-hop environment. The Boeing software shall enable more efficient operation (bandwidth utilization) over wireless networks without significantly reducing the successful routing of packets.
- **Quality-of-Service-based routing through DSCP-based link metrics.** The Composite Networking program managed by Dr. Clifford Warner of the SPAWAR Systems Center is developing a Next-Generation C2P that enables range extension over the Navy ADNS by using differentiated services codepoints (DSCP) to mark traffic. In the current OSPFv2 networks, special handling of DSCP-marked packets can only be accomplished through static configuration of router schedulers. By enabling DSCP-based link metrics for OSPFv2, DSCP-specific routes through the network can be constructed by the routing protocol

The implementation is based on modifications to the OSPFv2 reference implementation maintained by the OSPFv2 author, John Moy.[1] The software, called `ospfd`, is an open-source software implementation covered under the GNU General Public License (GPL). It supports both unicast and multicast (MOSPF) protocol operation The software is written in C++ and can operate as a user-space process using standardized kernel APIs.

The hardware platforms used for the demonstrations were Intel-based rack mount systems, ATX-chassis computers, and laptops, and ARM-based PDAs, all running Linux kernels in the 2.4 series. The computers used Ethernet, serial (RS422), and wireless LAN (802.11b) interfaces. Routers had an SNMP agent visible from standard SNMP monitoring tools.

4.2 Requirements

The following subsections summarize the program requirements. We have met all program requirements regarding implementation, integration, and testing, except those obviated by the selection of OSPFv2 extensions as the routing protocol to be developed under this contract.

4.2.1 Implementation requirements

The following implementation requirements are copied from [1]:

I-R1. The implementation shall be based on Linux 2.4 kernel.

I-R2. The implementation shall be written to operate as a user-space process, using standardized kernel APIs as much as possible.

I-R3. The implementation shall be written in the ANSI standard C or C++ programming language, compilable by the Gnu C compiler (gcc) version 2.96 or greater.

I-R4. The implementation shall be compatible with Ethernet (10BaseT and 100BaseT) interfaces

I-R5. The implementation shall be compatible with a serial interface (RS422).

I-R6. The implementation shall include an object that can be managed according to the Simple Network Management Protocol version 2 (SNMPv2) Network Management Framework

I-R7. The implementation shall log errors or unusual events to a system log file.

I-R8. The contractor shall prepare a "User's Guide" for the implementation, including how to configure and execute the implementation via a command-line interface and text configuration file, as well as configuration for system start up

[1] http://www.ospf.org

I-R9 The source code shall be delivered on CD-ROM media.

I-R10. The source code shall be coded and documented according to contractor's best current practices.

4.2.2 System integration and testing requirements

The following system integration and testing requirements are copied from [1]:

S-R1. The contractor shall install the implementation into a standard 1 or 2 rack-unit server including 10/100BaseT and serial PCI interface cards.

S-R2. The contractor shall install the implementation into standard mid-tower, ATX-based chassis should an insufficient number of rack-mountable servers be available for demonstration.

S-R3. The contractor shall test the correct operation of the serial and Ethernet interfaces.

S-R4. The implementation shall interoperate with routers based on OSPFv2 routing protocol via route summarization. **(Note: This requirement has been overcome by the decision to develop OSPFv2 extensions rather than a separate protocol. Therefore, there is no requirement to interwork via route summarization; interworking occurs through normal OSPFv2 mechanisms.)**

S-R5. The implementation shall interwork with available Linux traffic control software that supports prioritized handling of traffic based on quality-of-service markings in the packet header.

S-R6. The implementation shall be able to simultaneously coexist with OSPFv2 routing processes operating on other interfaces. **(Note: This requirement has been overcome by the decision to develop OSPFv2 extensions rather than a separate protocol. Therefore, there is no requirement to support multiple routing processes– a single OSPFv2 routing process shall support all interfaces.)**

S-R7. The contractor shall develop a test-suite to test the correct operation of the implementation. This test suite shall include link impairments designed to test the operation of the routing protocol.

S-R8. The contractor shall test the ability for a standard SNMPv2-based element manager to manage the implementation.

S-R9. The contractor shall define and conduct tests that ensure that the implementation is prepared for system-level experiments and demonstrations.

4.3 Software architecture

Running the user-space routing daemon, `ospfd`, turns a Linux machine into an OSPF router that communicates with the kernel to manipulate routing tables and send and receive packets with other OSPF routers. The implementation requires changes to allow for the new wireless interface type and for performing QOS-based routing. Figure 4-1 depicts the flow of data through the application. Numbers indicate points where the code has been modified for the NBN Composite Routing program to support DSCP-based metrics.

The Linux daemon interacts with the operating system kernel in several ways. The code for these standard Linux system calls has been modularized into a separate library, to ease portability to other platforms. For sending and receiving OSPF packets, the `sendmsg()`, `sendto()`, `select()`, `recvfrom()`, and `recvmsg()` calls are used with raw IP sockets. "Netlink" sockets handle the adding and deleting of routes in the kernel routing table and receiving interface information. The `signal()`, `sigaddset()`, and `setitimer()` kernel signaling functions manage timing events and shutdown. Logging is performed with the `syslog()` facility. The routing daemon is configured with a plaintext configuration file `/etc/ospfd.conf`; this file is parsed on startup using a TCL script.

4.3.1 Changes to support DSCP-enabled routing

Here is a brief overview of the modifications highlighted in Figure 4-1. We modified the configuration script (1) so that, within the configuration file, DSCP type metrics can be specified for each interface. The representation of the OSPF interface has an associated metric, which has been enlarged (2) to an array of metrics, one for each possible DSCP codepoint. When LSAs are originated by an interface (3), the appropriate DSCP link metrics are included when available. Each link (between two routers) is represented by a link object in the OSPF link-state database; we modified this data structure so that the cost of a link is an array that can store any type of cost indexed by DSCP codepoint (4). When incoming LSAs are parsed (5), the DSCP metrics are stored in the link object. The

routing calculation (6) is repeated for every DSCP type that has been encountered in the link-state database. Routes are stored in a routing table (7) based on their DSCP type. When calculating a route to a destination for a specific DSCP codepoint, and an end-to-end path using only DSCP-enabled metrics for that codepoint does not exist, a path will be constructed, if possible, based on all available type 0 (normal) links and metrics. Finally, unicast routes added to or deleted from the kernel (8) now carry a DSCP codepoint, and the kernel checks for a route based on the DSCP codepoint, selecting the type 0 route if no DSCP route is available.

Multicast routing in the Linux kernel is implemented via a multicast cache which presently does not support DSCP values in the same way as the unicast routing. However, we can still accomplish multicast routing based on DSCP codepoints without kernel modification, if we assume that all traffic to a particular multicast group carries the same DSCP codepoint. This is accomplished without kernel modification by having the MOSPF process examine the data packet for a DSCP codepoint and by then using the DSCP routing information, if available. The resultant multicast route installed in the kernel will be used by every multicast packet to that group, regardless of its DSCP codepoint.

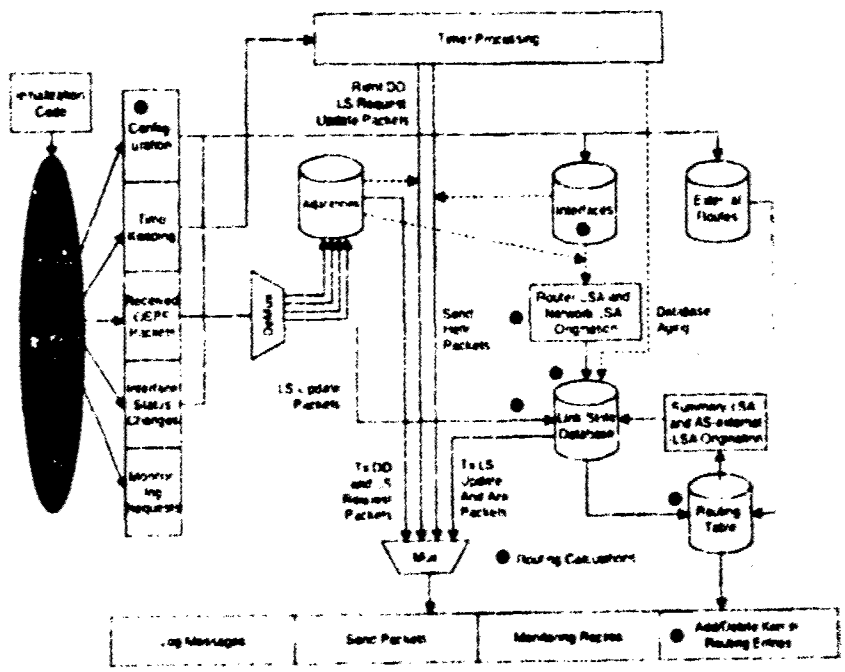


Figure 4-1 Software architecture for OSPFv2 daemon with DSCP modifications

The Linux kernel can make routing decisions based on the second octet of the IP header. In standard Linux kernels, this field is treated as the TOS value (defined by RFC 1349), which means that only three bits are used, allowing only eight possible values to differentiate routes. We use a patch for the kernel so this field is treated as a DSCP codepoint (defined by RFC 2472), allowing full six bits or 64 possible DSCP values. The patch works by redefining the mask that is applied to the second octet when the value is extracted from the IP header.

4.3.2 Implementation changes to support wireless interface type

We modified the configuration script (1) so that within the configuration file, wireless OSPF interface parameters could be specified, such as the wireless hello interval and flooding interval (LSF interval). In (2), a new interface was added for wireless OSPF. The interface is a hybrid version of the broadcast and point-to-multipoint interfaces. The need for adjacencies (3) is eliminated because we are using unreliable flooding, so database synchronization is no longer performed. Additional timers (4) have been added for the wireless hello interval, LSF interval, and

various associated expiration times are monitored. The origination of new LSAs (5) has been changed to occur on a periodic basis corresponding to the LSF interval as opposed to a reactive approach to link failure. The sending and flooding (6) of LSAs has been modified to enable the sending of wireless hellos and LSFs. In addition, the Multipoint Relay (MPR) algorithm is used to optimize the flooding of packets. Code to receive (7) the new packet types has been added. The use of database description, LSUs, LSAcks, and LSRs (8) has been eliminated on the wireless interface. Hello packets (9) are replaced with the wireless hello packet that contains a list of MPRs.

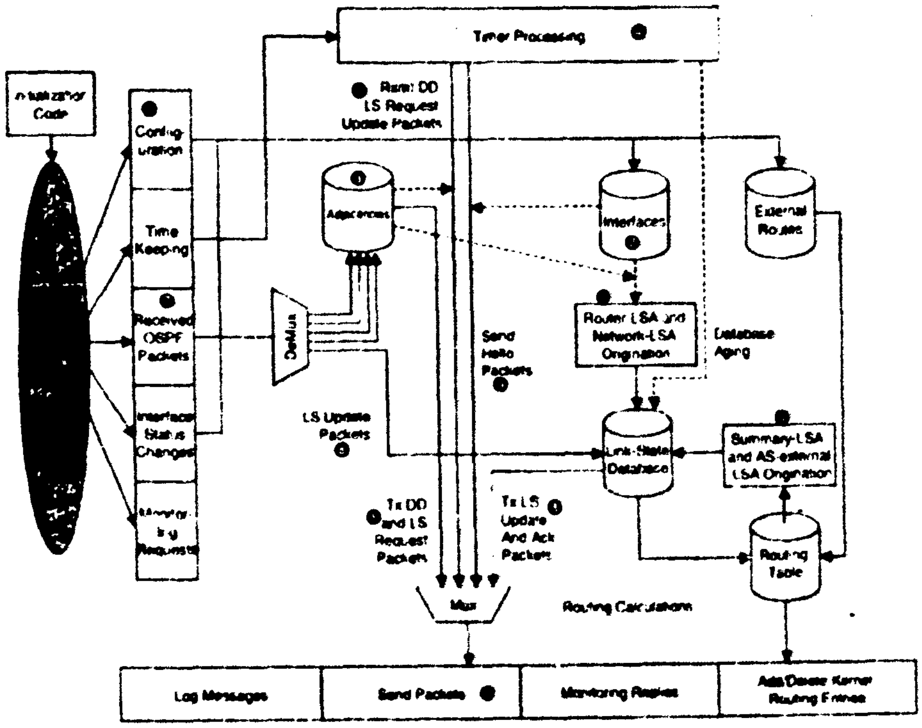


Figure 4-2. Software architecture for OSPFv2 daemon with wireless modifications

4.3.3 Event logging

In addition to the existing event logging that takes place when the daemon is run, the following items are now recorded to the log file (/var/log/ospfd.log):

- Log message that was "Received Pkt type 6" is now "Received Wireless Hello".
- Log message that was "Received Pkt type 7" is now "Received Link State Flood".
- When there is a link failure on the WANIC Interface's serial line
- When there is an IWSPY status change due to a bad signal-to-noise ratio (optional feature).
- When neighbors are added to or deleted from the IWSPY monitoring list (optional feature).
- If a DSCP MOSPF calculation fails, and uses the DSCP 0 multicast tree instead.
- If we substitute DSCP 0 metric for legacy router.
- When a new network LSA is received on a wireless interface that obsoletes an existing one.
- When a new default route is added or deleted for a wireless stub area.
- When the MOSPF cache is cleared, to better understand when multicast cache entries are absent.
- When a multicast routing calculation is requested for an invalid multicast source.

- After each dijkstra calculation, the number of dijkstra calculations that have occurred since the daemon was started. (This statistic is incremented once for each recalculation, not once per DSCP dijkstra run.)

An option may be turned on at compile-time to generate a second log file (/var/log/ospfd.log2). This additional log file is helpful for keeping track of changes to the wireless network. The following items are logged when a wireless interface type is defined:

- The list of one-hop neighbors, and the corresponding two-hop neighbors that can be reached through them.
- The list of neighbors this node has selected as MPRs (MPR list).
- The list of neighbors selecting this node as MPR (MPR selector list).
- When an LSU is being forwarded.

4.4 Performance metrics/test criteria

The OSPFv2 software was evaluated on the basis of interoperability, correctness, and performance. With respect to interoperability, the Linux-based OSPFv2 routers were connected to a network containing up to three Cisco routers running OSPFv2, and tests were performed demonstrating the modified OSPFv2 software successfully interoperating with the Cisco software. The correctness of the implementation was tested by configuring the testbed to exercise certain parts of the code and by observing the resulting routing tables and packet traces from experiments. The software was also tested for performance by specifically contrasting the operation of OSPFv2 legacy software with the operation of the Boeing extensions, using selected network configurations. Traffic generation and post processing of traffic traces was used to establish performance metrics, including the following key metrics:

i) OSPFv2 overhead consumed (measured at IP layer);

ii) Packet delivery ratio for user data traffic, and

iii) Convergence time of routing tables

4.5 Equipment/software to be used

In addition to the Linux based routers, our testing used the equipment and software listed below

4.5.1 Routers

Cisco 3600 series: The Cisco 3600 series router is a modular, multiservice access platform for medium and large-sized offices and smaller Internet Service Providers. Specifically, our laboratory testing used Cisco 3640s equipped with IOS version 12.2-4

Cisco MAR series: Our demonstration used Cisco Mobile Access Routers (MAR), a special PC-104 form factor router capable of RIPv2 and OSPFv2 routing

OLSR routing software: We have modified the OLSR routing software distributed by Joe Macker (NRL) for operation with associated stub subnets. We used this software to compare its performance against that of our modified OSPFv2 implementation

4.5.2 Interface cards

Most routers were equipped with PCI-based 10/100 Ethernet cards, typically Intel EtherExpress Pro. In addition, we tested with 802.11b cards such as Orinoco Gold cards and access points, and we used the Wanic 522 serial card to ensure serial interface compatibility

4.5.3 Data channel simulators

We used the following channel simulators and emulators

MobiEmu: MobiEmu is a tool for emulating mobile ad hoc network environment with a fixed network of Linux machines (kernel 2.4 and above) in a Lab setting. It can support virtually any mobility scenario (such as ns2

scenarios) without actually moving the nodes physically. It is good for MANET (mobile ad-hoc network) research that requires testing a real implementation or application with different mobility patterns.

Adtech SX/13: The SX Data Link Simulator creates the same delay and error characteristics caused by long distance terrestrial and satellite data links. With dual-channel, full-duplex operation, the simulator provides bi directional testing with programmable delays, random bit errors and burst errors. Multiple delay and error events can be programmed to simulate a wide variety of chronic and periodic conditions or events such as peak traffic times and equipment overloads. In our testing, the Adtech SX/13 will be used to as channel simulators for the serial interfaces

4.5.4 Traffic generation and measurement

Chariot: Chariot is a commercial tool that emulates network users by sending traffic over the network. It emulates multiple data types and rates using different protocols, and measures performance between pairs of networked computers. All computers included in tests have NetIQ Performance software installed. Chariot uses real data to emulate and test different applications through the network, and to monitor from its console all traffic between endpoints We used Chariot at our interim demonstration in May, 2003.

MGEN/DREC: The Multi-Generator (MGEN) is open source software by the Naval Research Laboratory (NRL). MGEN provides the ability to perform IP network performance tests and measurements using UDP/IP traffic (TCP is currently being developed) The toolset generates real-time traffic patterns so that the network can be loaded in a variety of ways. The generated traffic can also be received and logged for analyses. Script files are used to drive the generated loading patterns over the course of time. These script files can be used to emulate the traffic patterns of unicast and/or multicast UDP/IP applications. The receive portion of this tool set can be scripted to dynamically join and leave IP multicast groups. MGEN log data can be used to calculate performance statistics on throughput, packet loss rates, communication delay, and more.

Tcpdump: Tcpdump is a packet-capturing utility for Unix-based machines. It allows a user to view the live network traffic passing on the wire by/through the user's host. It displays output on a console or can write binary files into a standard format for post-processing.

Ethereal: Ethereal is a free network protocol analyzer for Unix and Windows, which provides a nice graphical user interface for tcpdump Ethereal has several powerful features, including a rich display filter language and the ability to view the reconstructed stream of a TCP session We plan to write extensions to Ethereal to support the OSPFv2 protocol extensions we are developing.

Custom logging scripts: Part of our testing required logging of information regarding routing table updates and packet delivery We developed customized scripts on an as-needed basis to support our data collection and analysis.

4.5.5 Network management software

scotty: The open-source scotty tool provides a GUI-based capability to monitor SNMP agents on resident devices

Net-SNMP agent: We installed the open-source net-SNMP agent on the routers.

4.5.6 QoS software

The Linux advanced routing and traffic control (LARTC) framework provides a number of built-in tools to manage traffic for QoS management. In our tests and demonstrations, we combined the DSCP-based routing extensions with DSCP-enabled priority queue scheduling

4.5.7 Software verification

Insure++: Insure++ is a commercially-available run-time C code analyzer that is designed to examine code for many classes of implementation errors not commonly found by compilers, including memory access violations, memory leaks, pointer errors, data type errors, and library errors.

5 Wireless OSPFv2 testing

Our two main objectives in wireless OSPFv2 testing were to (i) validate that the basic operation of the protocol, within a single MANET subnet, was similar to that observed in simulation, and (ii) test the implementation for correctness in a variety of network configurations

5.1 OSPFv2 router configuration

Figure 5-1 illustrates a representative OSPFv2 configuration file. Unless otherwise specified, the values shown in Figure 5-1 were used in the tests described below

```
routerid 10.0.0.2            /* Variable                                        */
mospf                        /* Enables MOSPF                                   */
dscp routing                 /* Enables DSCP-based routing                      */
log_level 0                  /* Extent of log messages found in
                              *  /var/log/ospfd.log                             */
#
area 0.0.0.0                 /* classify the follow interfaces in area 0        */
interface 10.0.0.2 1         /* assign interface and metric                     */
wospf                        /* set the interface type as wospf                 */
#StubWireless                /* optimize interface by setting as a stub         */
ospfIfAuthType 2
md5key 1 sharedsecret        /* set up the use of authentication
                              * with md5sum                                     */
ospfIfWHelloInterval 2       /* set the hello interval to 2 seconds             */
ospfIfLsfInterval 2          /* set the LSF interval to 2 seconds               */
mpr time 6                   /* set the MPR Selector expiration interval        */
ospfIfPtrWDeadInterval 6     /* set the neighbor dead interval                  */
igmpQueryInterval 10         /* set the igmp query interval for faster
                              * multicast routing                               */
interface eth1 1             /* assign broadcast interface and metric           */
```

Figure 5-1 Default router configuration for wireless OSPF

5.2 Basic MANET testing

This test is a fundamental test of how the wireless OSPFv2 interface performs in comparison to an OLSR implementation and a legacy Point-to-Multipoint (PTMP) OSPF implementation. In addition, we wrote scripts that allowed us to exactly replicate the wireless emulation environment in simulation, and compared our wireless OSPFv2 implementation with our simulation models that were used in [7].

This testing consisted of N routers with Ethernet interfaces connected to the mobile network emulator as shown in Figure 5.2-1

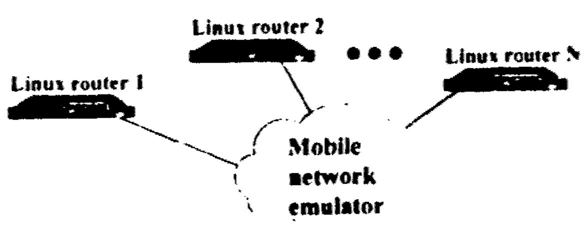


Figure 5.2-1 Basic MANET Routing Testbed

Test procedures: The synthetic network emulator (SNE) was configured with three scripts of increasing mobility. The mobility area was 500m x 500m and the simulation ran for 30 minutes. The radio range was 240m in the 8 node case and 200m in the 16 node cast. The max speed and max pause time for the high, medium, and low scenarios were 100m/s and 1s, 25m/s and 4s, and 6.25m/s and 16s respectively. The Linux routers were first

configured in point-to-multipoint mode, and baseline tests were run. Next, OLSR was installed on all routers and identical tests were conducted. Finally, the Linux routers were reconfigured in wireless OSPF interface mode and the tests were repeated again. The routing daemons were started asynchronously over an initial 30 minute startup period in order to avoid bursts of overhead when timers fire periodically. The hello, LSF, and TC intervals were each set to 2 seconds unless otherwise specified below, and the dead intervals were set to three or four times the hello interval. Each node sent a 16 byte UDP packet to every other node on 5 second intervals so statistics could be calculated on the packet delivery ratio.

Measurements and data processing: We ran tcpdump on each router and saved the results to a log file. The logfile contained the UDP traffic sent and received in addition to the overhead generated by OSPF. We then post-processed the data to extract utilization and delivery ratio statistics. Network routing tables were periodically dumped and post-processed, and compared against the network emulation scenario.

5.2.1 Implementation results

The results below (Figures 5.2-2 through 5.2-5) give the overhead and delivery ratio for 8 nodes and then 16 nodes. In the 8 node case, standard OSPF PTMP is in a range where it functions as good or better than OLSR and WOSPF. The 8 node scenario does not portray the adverse affects of large amounts of overhead. However, when the 8 node scenario is taken in context with the 16 node scenario it is evident that PTMP OSPF cannot be scaled much over 16 nodes. In addition, it is seen that the overhead produced by PTMP is significantly impacted by the degree of mobility because routing is triggered by link changes. The overhead produced by OLSR and WOSPF is not significantly impacted by mobility because routing is triggered by periodic timers. However, the overhead is directly affected by the timer intervals. For less mobile scenarios, timer intervals can be significantly extended above the 2 second interval tested here without significantly compromising routing protocol performance

OLSR yields good results in the 8 and 16 node scenarios. The overhead is at a minimal level while the delivery ratio is comparable to OSPF PTMP. This result is to be expected because OLSR is optimized for the basic MANET network. It is seen that WOSPF scales much better with respect to overhead than PTMP. This scaling property is the main advantage in using a wireless interface.

WOSPF generates approximately 2 or 3 times more overhead than OLSR. The increase in overhead is because OSPF is a full link state protocol, with each node originating an LSA. In contrast, OLSR only generates TC messages from those routers who are MPRs. This causes WOSPF to have a slightly higher delivery ratio, but it generates more overhead in well connected networks. WOSPF does not adopt the OLSR TC message mechanism because it represents more of a departure from how OSPF implements its databases, thereby affecting compatibility with legacy OSPF routers.

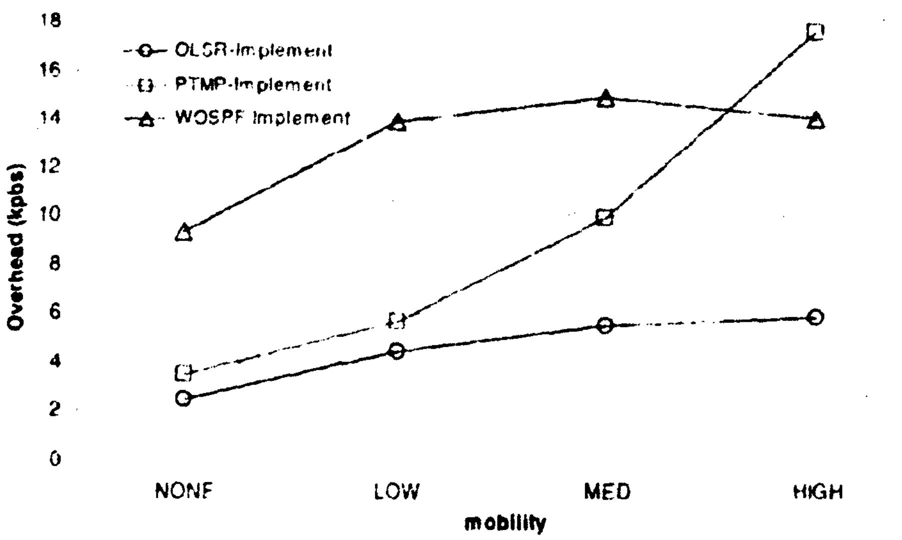


Figure 5.2-2 Comparison of the overhead generated by implemented protocols in an 8 node network

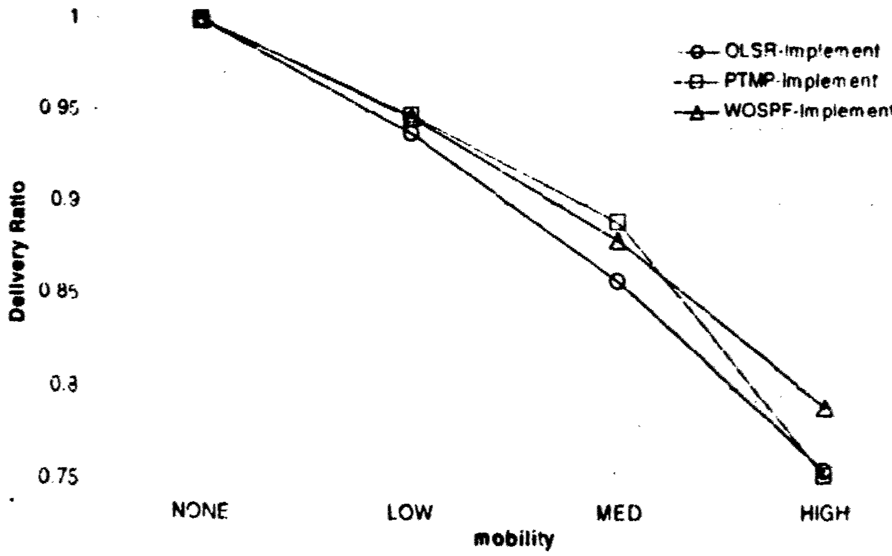


Figure 5.2.3 Comparison of the delivery ratio obtained from implemented protocols in an 8 node network

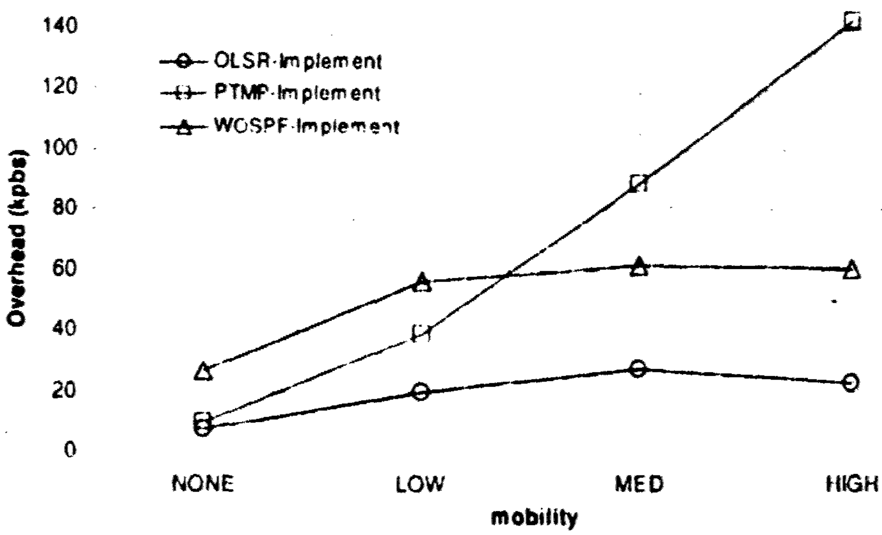


Figure 5.2-4 Comparison of the overhead generated by implemented protocols in a 16 node network.

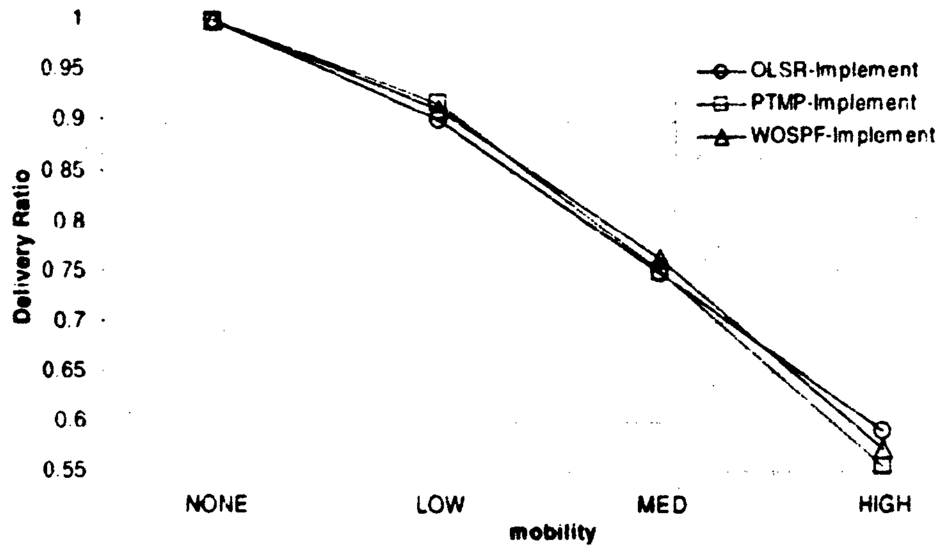


Figure 5.2-5 Comparison of the delivery ratio obtained from implemented protocols in a 16 node network.

5.2.2 Simulation comparison results

We next ported our wireless network emulation scripts to the QualNet simulator, so that we could reproduce the exact network behavior in simulation. This exercise allowed us to validate our simulation models. Figures 5.2-6 through 5.2-11 show the close match between simulation and implementation results shown in the previous graph. This close match gives us confidence that our simulation can be faithfully scaled upwards in the number of nodes.

Although the comparison in results is very close, in all three protocols, the delivery ratio is, on average, slightly lower for the implementation. In addition, for the eight node case, the overhead is lower for implementation when at the highest mobility. This is probably due to slight differences in the QualNet and implementation mobility scripts.

The results found in simulation for the PTMP OSPF compare very closely with those found in simulation. There is a minor discrepancy in delivery ratio, but this could be due to minor variations in the mobility scenarios.

OLSR also yields similar results in simulation and implementation. There is some difference in the overhead that is caused by dissimilar OLSR versions. The implementation is version 3 while the QualNet protocol is version 7. The packet fields were modified, making version 7 more overhead intensive.

The comparison of implementation and QualNet is very similar for WOSPF.

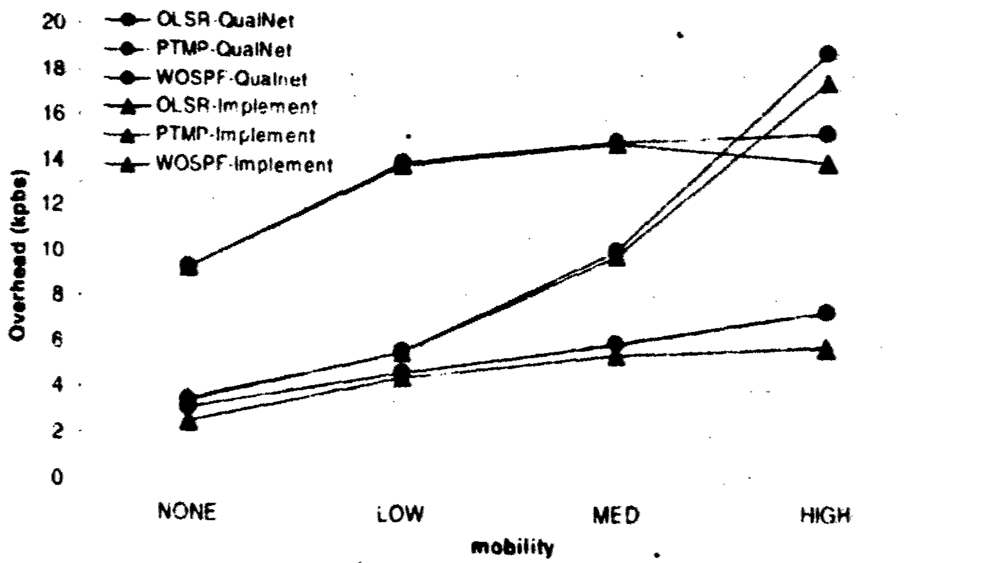


Figure 5.2-6 Comparison of the overhead generated by the implementation and QualNet in an 8 node network

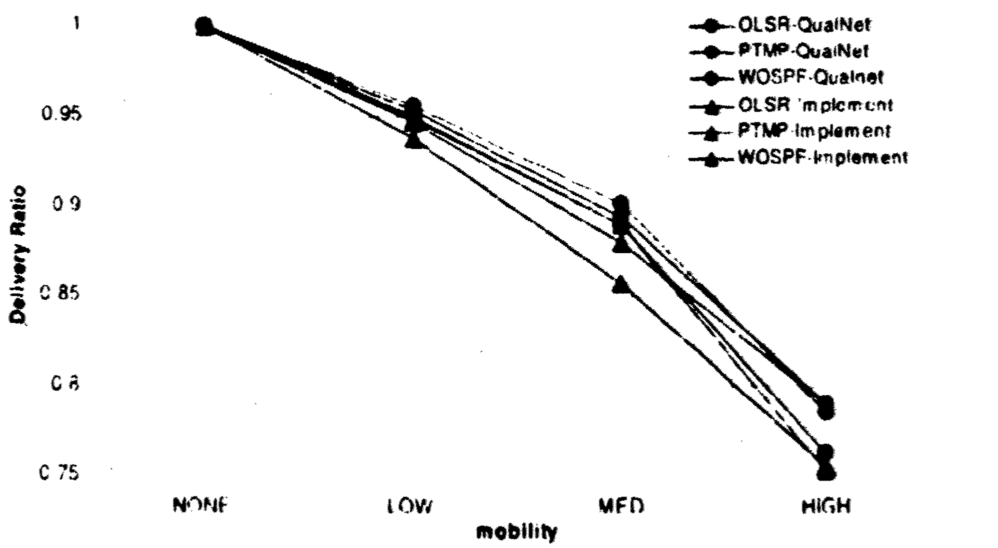


Figure 5.2-7 Comparison of the delivery ratio obtained by the implementation and QualNet in an 8 node network

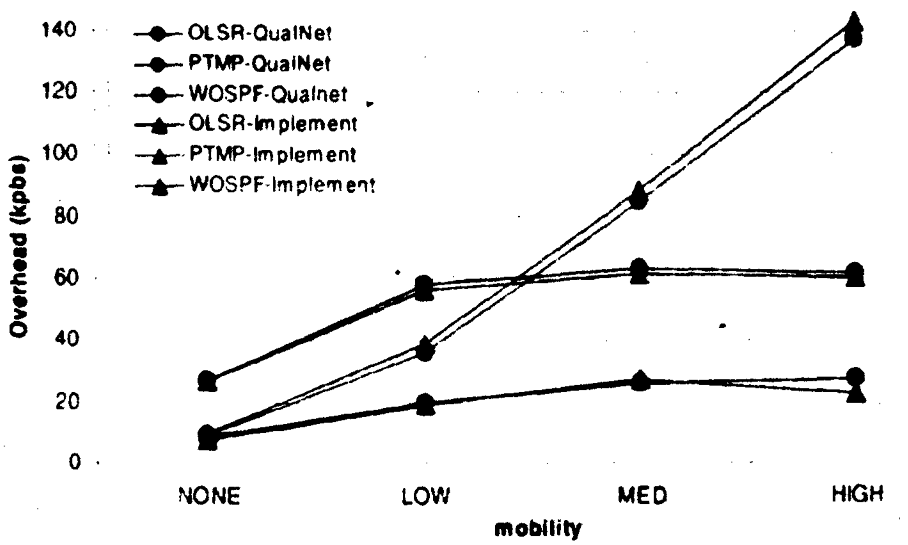


Figure 5.2-8 Comparison of the overhead generated by the implementation and QualNet in a 16 node network.

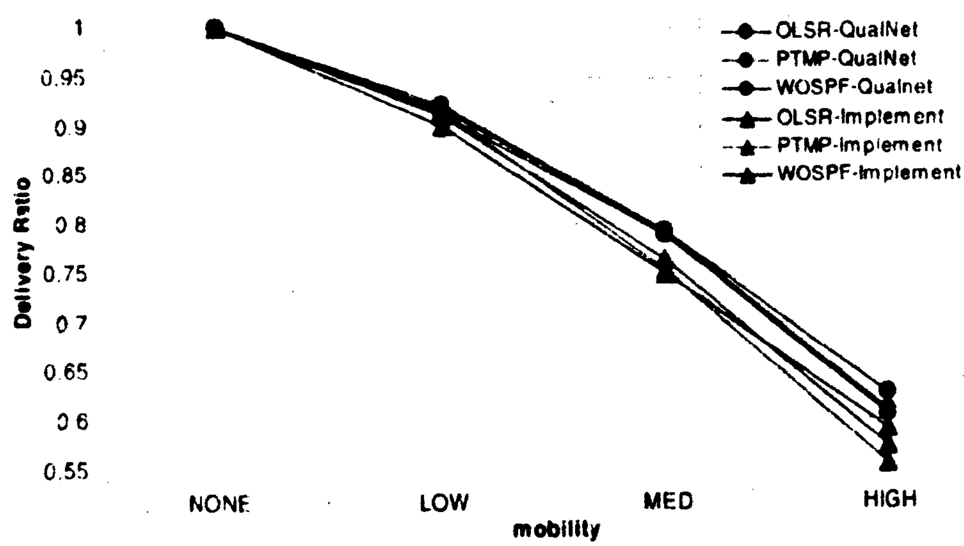


Figure 5.2-9 Comparison of the delivery ratio obtained by the implementation and QualNet in a 16 node network

From the results shown in Figures 5.2-5 to 5.2-9, it is observed mobility has little impact on the overhead produced by WOSPF, but mobility highly effects the overhead produced by PTMP with 2 second timers. The results shown in Figures 5.2-10 to 5.2-11 show the similar tests but with slower mobility and 10 second timers. The mobility classified as "LOW" in the previous figures was considered as "HIGH" mobility in the next two figures, and "LOW" and "MEDIUM" mobility rates were rescaled accordingly.

In these results, we again observe that WOSPF is more resistant to overhead swamping due to mobility than PTMP. When the network is highly mobile (approximately a 70% delivery ratio), the overhead of OSPF PTMP is 4 to 5 times greater than WOSPF. Another interesting observation is that there is more overhead with the 10 second timer at "HIGH" mobility (same as "LOW" mobility with 2 second timer) than the 2 second timer at "LOW" mobility, again, these mobility rates are the same. The results indicate that extra hello traffic in PTMP can help delay the swamping of a network because links are changing faster than routers can synchronize.

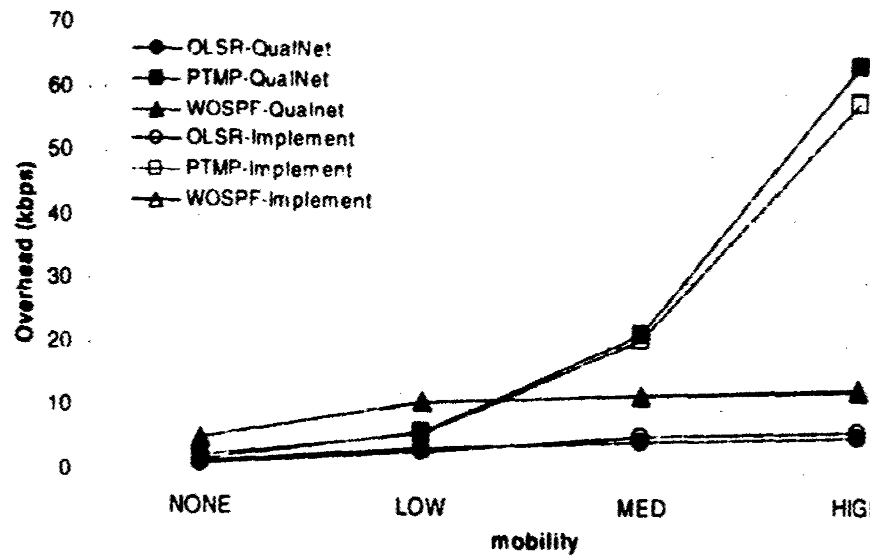


Figure 5.2-10 Comparison of the overhead generated by the implementation and QualNet in a 16 node network with timers set on 10 second intervals

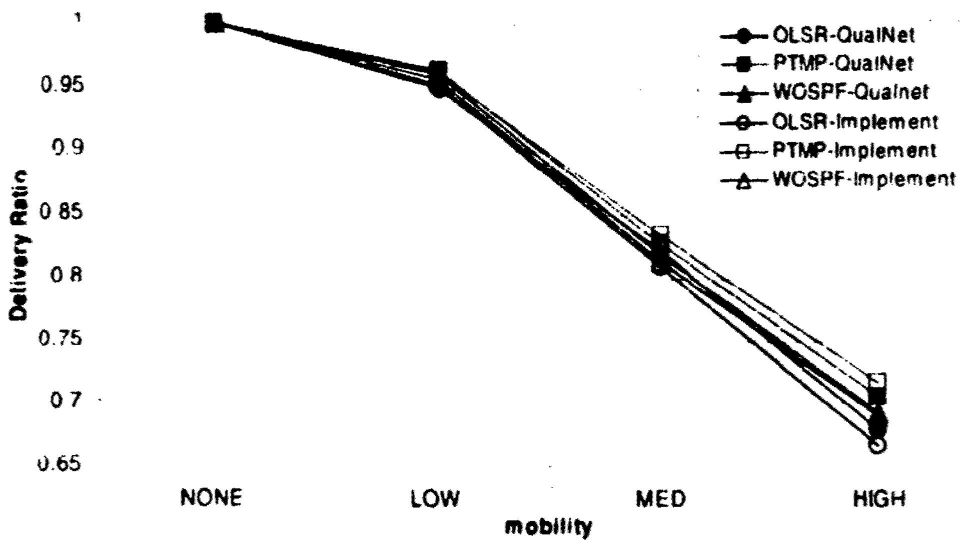


Figure 5.2-11 Comparison of the delivery ratio obtained by the implementation and QualNet in a 16 node network with timers set on 10 second intervals

The amount of overhead generated by PTMP in 8 and 16 node networks is manageable in 802.11 networks running at 11Mbps. However, the above results show the trend that PTMP does not scale well with respect to the number of routers in a network. The following results (simulation only) show how PTMP scales with respect to WOSPF for between 8 and 36 nodes.

All previous results were obtained for comparison validating simulation with implementation, so Ethernet was used at the link layer (the wireless network emulator is actually an Ethernet bridge which selectively enables and disables reception of MAC addresses, but the underlying CSMA/CD MAC properties are in effect). Use of an Ethernet link layer implies that the collisions were very minimal and the transmission time was low. In the following results, the MAC layer used was 802.11b with RTS/CTS enabled. Each router has a radio range of 250

meters. The mobility of each node was set to move to a random location between 0 and 10m/s and then pause for 40s before moving again.

The data below comes in two forms for PTMP and WOSPF. One line (solid shapes) shows the results when the node density is scaled. This means the simulation area remains the same and the number of nodes increases. The second line (hollow shapes) shows a static node density. The simulation is scaled with respect to the number of nodes according to $d = \sqrt{\frac{500^2 N}{16}}$, where N = number of nodes. All results for scaled node density were run in a 500m by 500m area. The static node density scenarios were run with a node density of 15,625 m^2/node (corresponding to 16 nodes in a 500m x 500m grid).

Figures 5.2-12 and 5.2-13 show that WOSPF scales much better than PTMP as the number of nodes increases. Also, we see that PTMP scales better when the network is less dense. When the node density is kept static (the simulation area is scaled) as the number of nodes increases, the overhead is almost half as much at 36 nodes as when the node density is scaled (simulation area is kept static). This is because there are fewer adjacencies to be maintained in a sparse network.

The opposite is true for WOSPF, however, it is difficult to see the difference with the figure scale. Here the overhead increases when the network is sparse because almost every node must be an MPR. So, optimized flooding becomes classic broadcast flooding. When a network is dense most traffic is flooded only once.

The delivery ratio, shown in Figure 5.2.13, with scaled node density stays about the same because the number of hops to reach a node stays about the same. In the static node density scenario, the number of hops consistently increases as the simulation area increases. The increase in area leads to the possibility of more partitioned nodes or bad routes. The PTMP delivery ratio approaches zero after the overhead reaches a point of swamping the network. Here we see a key problem with PTMPs method of exchanging routing information.

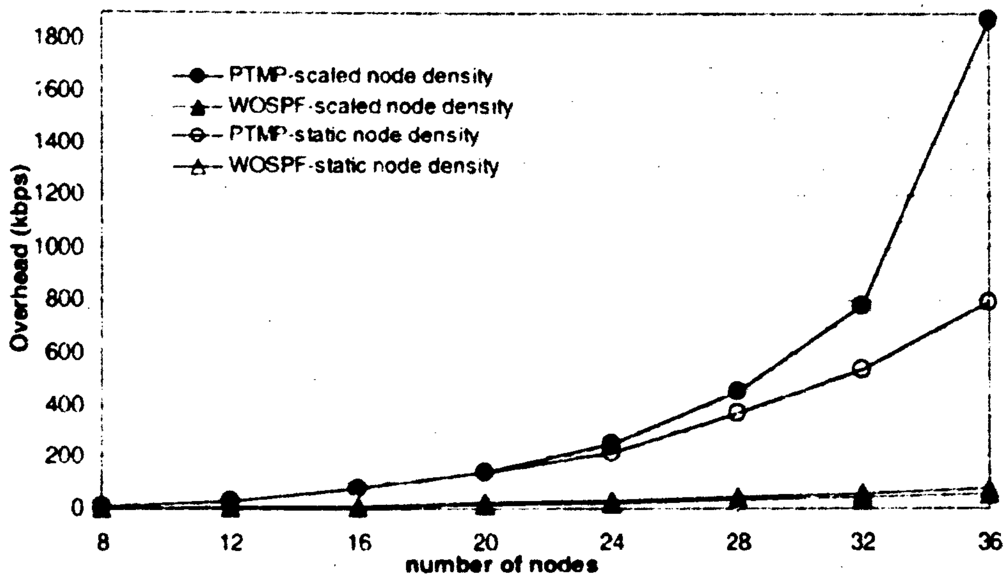


Figure 5.2-12 Overhead obtained in QualNet when scaling the number of routers in the network, with timers set on 10 second intervals

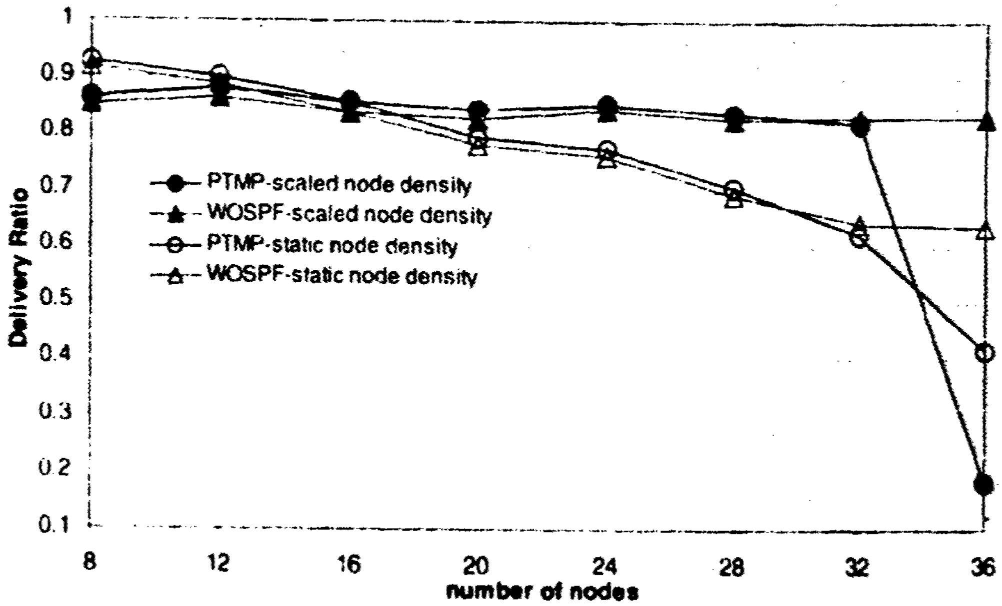


Figure 5.2-13 Overhead obtained in QualNet when scaling the number of routers in the network, with timers set on 10 second intervals

5.3 Other wireless OSPFv2 tests

The tests described in Section 5.2 were used to characterize performance. The following set of tests was performed to validate the correct operation of the implementation.

Test name: Heterogeneous Network, Single Area

Objectives: Confirm the operation of edge (routers with more than one interface with one being wireless) routers, the correct handling of heterogeneous links, and routing in a multi-path environment in a single OSPFv2 area.

Equipment used: As shown in Figure 5.3-1, 14 routers with Ethernet interfaces are connected to the mobile network emulator, and two of these routers are further connected to routers A and B respectively using a cross over Ethernet cable. Routers A and B are interconnected by an Ethernet cable that is bandwidth limited by Linux Traffic Control (TC). Two end users (source and sink) generate UDP traffic and are attached to routers A and B.

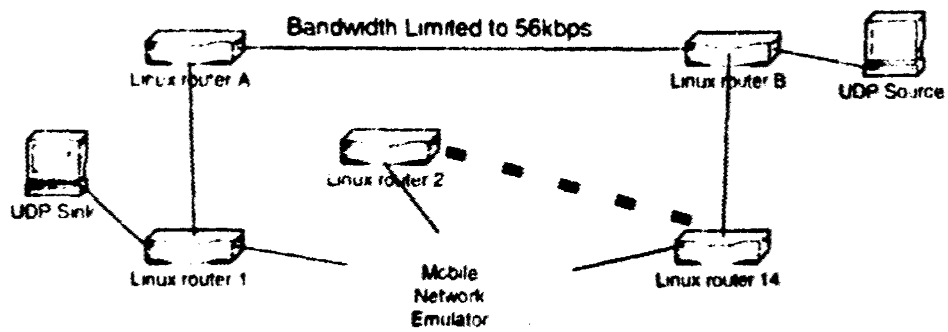


Figure 5.3-1 Heterogeneous Network

Test procedures: Linux TC traffic control limits the data rate between routers A and B to 56kbps. UDP traffic was sent from sink to source at 128kps. OSPFv2 was used on each router using the wireless interfaces on the mobile network. The low bandwidth link was assigned a link metric of 100 and all other links a cost of 1.

Three tests were run and the delivery ratio was calculated. The first test consisted of partitioning routers 1 and 14, so all traffic was forced over the low bandwidth link. Second, the low bandwidth link was brought down and a multi-hop path was created on the wireless network. Third, the network emulator was given a mobility script where node 1 and 14 would be partitioned at times and low bandwidth was in the up state.

Measurements, data processing, and outcome: The edge routers appropriately handled the inclusion of point-to-point links. OSPFv2 directed most traffic through the high bandwidth mobile network when it was available. Figure 5.3-2 illustrates the user data packet delivery ratios. In test 1, the low bandwidth link could only handle half of the load, so the delivery ratio was 49%. In test 2, the delivery ratio was 100% because OSPF was able to route through the wireless network, which was only partially loaded. A protocol such as OLSR that cannot assign link metrics, directs traffic through the low bandwidth point-to-point because it is the shortest path. This would lead to the delivery ratio found in test 1. Finally in test 3, the delivery ratio was 82%. Traffic went over the wireless network unless there was a network partition. Packets were either dropped due to re-routing time or queue drops over the low bandwidth link.

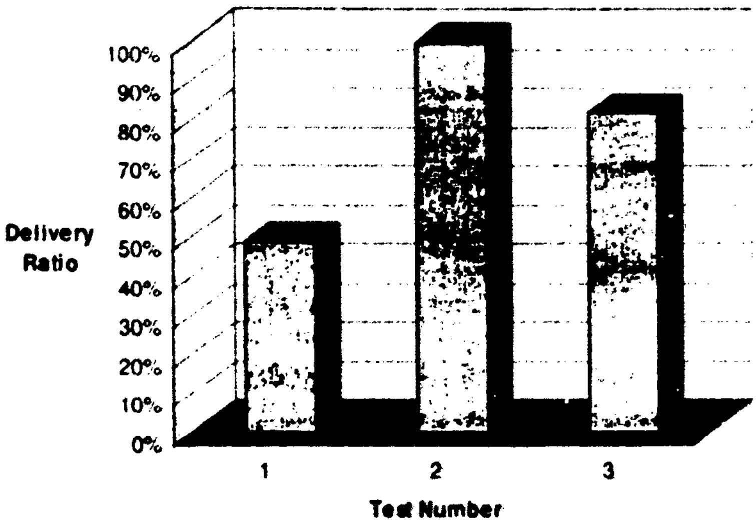


Figure 5.3-2 Delivery Ratios for the three Heterogeneous Network tests.

Test name: Multi-area operation using Wireless OSPFv2

Objectives: Confirm that wireless OSPFv2 network can operate in multi-area environment, and that Linux and Cisco routers are interoperable.

Equipment used: Sixteen Linux routers with Ethernet interfaces were connected to the network emulator, three Cisco routers were connected in series with the emulated routers as shown in Figure 5.3-3, and two end users were used as data sources and sinks. Each of the Linux routers were configured with the wireless OSPF interface type on the emulator. Routers 1 and 16 were additionally configured with broadcast interfaces. All Linux routers interfaces were set to Area 1.

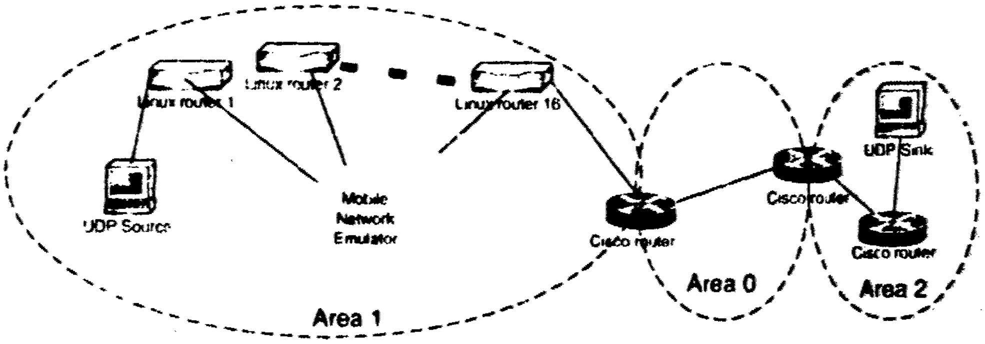


Figure 5.3-3 Multi-area operation with wireless network.3

Test procedures: Routing tables and LSA databases were checked to see if routing was being done correctly. Traffic was sent from source to sink. The Mobility emulator was run for a random mobility script to see if rerouting was done correctly.

Measurements, data processing, and outcome:

Wireless OSPF was shown to work properly in a multi-area environment, and it interoperated with Cisco routers. All routing tables and LSA database were correct. Traffic was successfully routed from source to sink and rerouted when the wireless network changed.

Test name: Stub LSF with multi-area and multi-AS (Autonomous System)

Objectives: Show that wireless OSPF can be used in a multi-AS network, and that wireless OSPF can be used in a stub wireless format that reduces routing overhead.

Equipment used: Same as in the multi area test without the UDP source (Figure 5.3-4).

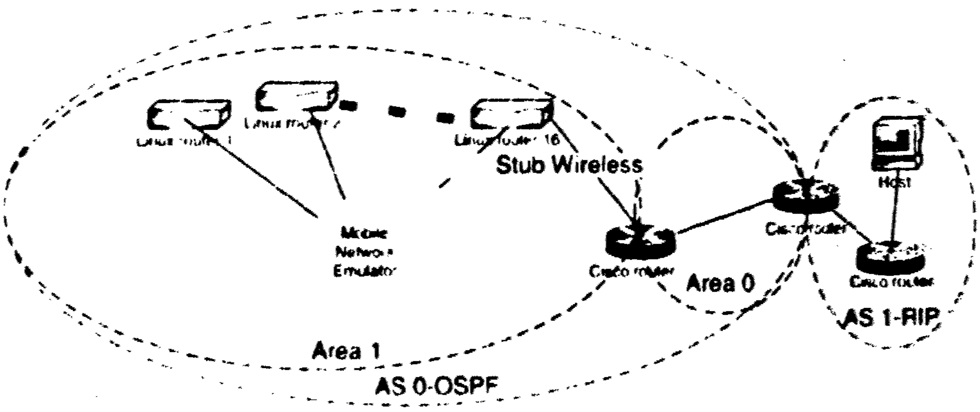


Figure 5.3-4 Multi-AS operation with wireless network

Test procedures: The network is setup the same as the previous test. Area 1, which is the wireless network, and Area 0, consisting of two of the Cisco routers, comprise Autonomous System 0. Area 2 on the Cisco routers is brought down and replaced with the RIP routing protocol, to form another AS, Autonomous System 1. OSPF is redistributed into RIP, and RIP is redistributed into OSPF on the middle router, which serves as an Autonomous System Border Router (ASBR). The edge Linux router possessing both a wireless and wired interface is configured as the default router for the wireless network by setting the "StubWireless" parameter.

The test consisted of monitoring the LSA databases and routing tables in the routers for correctness. In a stub wireless network, only LSAs from routers in the wireless should be present in the LSA databases. Host was set to

ping Linux router 1 while the wireless network was in motion. The stub wireless network was checked for a default route to the outside, and the Cisco were checked for routes to nodes in the wireless network.

Measurements, data processing, and outcome:

Operation in a multi-AS network was successful. AS External LSAs were generated by the Cisco bordering the autonomous systems and the LSA were disseminated. Routers in the stub wireless network were successful in reducing overheat yet still maintaining connectivity to the outside by creating default routes to router 16. Each of the wireless routers, except router 16, contained only LSAs from within the wireless network so overhead was reduced. All routing tables were checked and were correct. The ping sent from Host was successfully sent, received, and rerouted when necessary.

Test name: Multiple interfaces (two wired, two wireless)

Objectives: Show that a Linux router running ospfd can run multiple wireless and wired interfaces on the same router.

Equipment used: Sixteen Linux routers are connected to the network emulator. Two of the Linux routers are also on an 802.11 wireless network. Additionally, there are point-to-point links as shown in Figure 5.3-5.

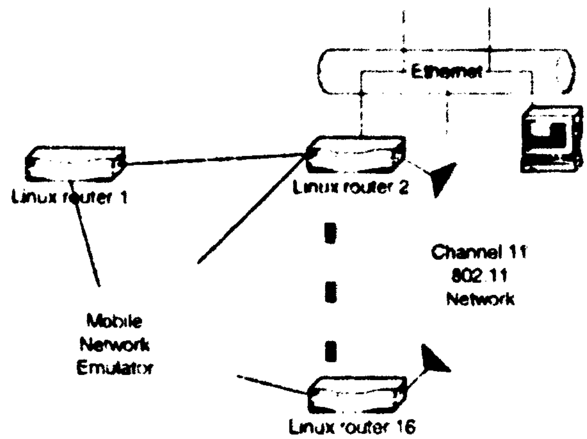


Figure 5.3-5 Ospfd router with two wireless interfaces and two wired interfaces

Test procedures: Check that Linux router 2 has the right information in its LSA database and that LSAs are distributed correctly. In addition, the point-to-point wired links were brought up and down and a network emulator mobility script was run. The routing table was checked as links were varied for correctness.

Measurements, data processing, and outcome:

Correct operation was verified for Linux router 2. All LSAs were generated and distributed correctly. As links were brought up and down, rerouting functioned properly.

Test name: AS opaque dissemination

Objectives: Show that ospfd with when running a wireless interface is able to disseminate opaque LSAs.

Equipment used: Sixteen Linux routers on the network emulator and two additional routers connected by Ethernet as shown in Figure 5.3-6. Zebra routing software version 0.93b is installed and configured on one of the Linux routers (shown in Figure 5.3-6)

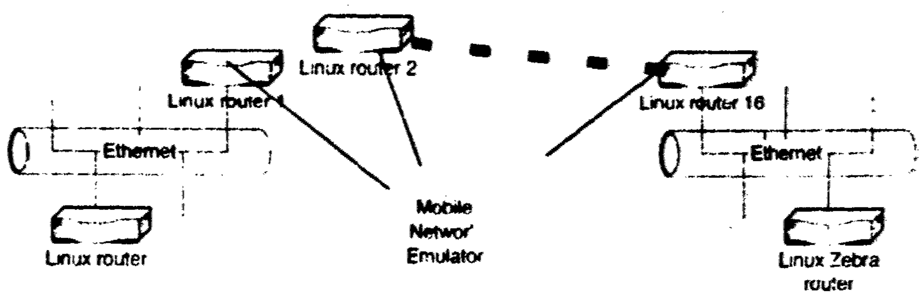


Figure 5.3-6 Opaque LSA dissemination

Test procedures: The Zebra OSPF daemon is configured to support MPLS-TE, Traffic Engineering extensions, which are disseminated via Type-10 Opaque LSAs. Each of the routers LSA databases will be checked to verify that the type 10 LSA is present.

Measurements, data processing, and outcome: The dissemination of Type-10 Opaque LSAs was verified throughout the entire wireless network.

5.4 Summary

This section has described our testing results for the wireless interface type. We obtained good agreement between implementation and simulation results. In addition, we observed the following characteristics when testing the implementation

- Multicast operation over 802.11 interfaces has several problems that could not be addressed in the scope of this contract. We list a few of the problems we discovered here. First, when a wireless node decides to join a group it issues an IGMP join on its interface. Every router in radio range that receives this message then creates a group membership LSA, so the router can distribute traffic to the sink. The IGMP mechanism was designed for wired networks in which many unnecessary routers do not receive the IGMP join. Second, multicast routeing entries were created for wired links. The routing entry gives input and output interfaces for a particular multicast group. When packets are handled like this on a wireless network, the routing entry input and output interfaces are the same. This leads to packets being caught in routing loops that replicate packets until they reach their max TTL. This will swamp a network unless TTL is set sufficiently low (this is a hack to make it work for small networks) or a cache is kept at each node that will only send the same packet once.
- Using multiple copies of vmware on one physical machine causes lag for each virtual machine. Access to the real time clock is available to only one of the virtual machines at a time, so time runs at a different rate on the other virtual machines. This causes time-dependant applications, such as the OSPF daemon, to behave slower than normal
- The capability to have multiple OSPF areas within the same wireless network has not been implemented, because the specification for inter-area wireless routing has not been defined. In all of the tests, the wireless network is treated as a single area, which is capable of connecting to other OSPF areas. A wireless edge router cannot at this time be an area border router
- To obtain statistics on link-layer errors in the physical layer of wireless 802.11b drivers on Linux, one must put the driver in a special monitoring mode. In monitoring mode, the driver can only receive network packets but not transmit, so it is not possible to utilize this capability while running OSPF. This limits the type of link-layer notifications that can be provided to OSPF to the signal-to-noise ratio of a link. The drivers can only monitor the radio noise level of a maximum of eight peers

We anticipate performing some additional development on the wireless interface type during the ONR KSA FNC Block 2 program. Specific areas of interest include optimizations for handling external LSAs, an acknowledgment mechanism for a subset of LSA flooding, and adaptive protocol mechanisms (link metrics, timer intervals, etc.). This work has been submitted to the IETF for consideration (draft-spagnolo-manet-ospf-wireless-interface-00.txt).

6 DSCP-based routing

Our two main objectives in DSCP-based routing testing were to (i) test the ability of the routers to build DSCP-specific routes, (ii) test the ability of the DSCP-based routing extension to interoperate with legacy routers, and (iii) test the interoperability of DSCP-based routing with DSCP-based scheduling.

6.1 Test results

Test name: Multiple DSCP paths

Objectives: Confirm the correct operation of unicast DSCP routing and that the proper route is created for each of four DSCP codepoints.

Equipment used: Nine Linux routers, two traffic sources and a traffic sink, and a timeserver to synchronize all the clocks used in this test. The network formation is shown in Figure 6-1.

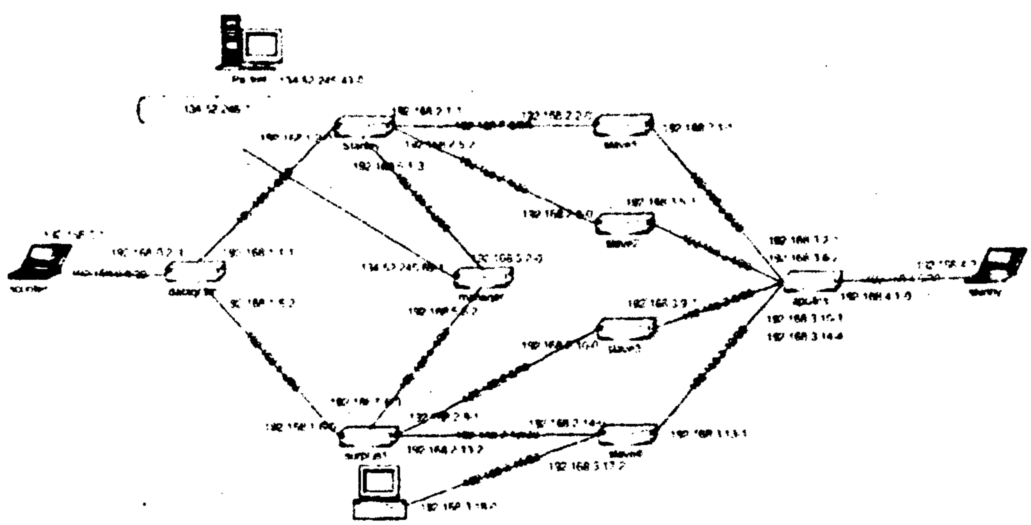


Figure 6-1 Multiple DSCP paths test configuration

Test procedures: Each of the routers were configured with broadcast interfaces. Link metrics were assigned costs according to the values found in Table 6-1. All router interfaces not found in the table were assigned a cost of one for all DSCP values. The cost assignments were set so traffic of DSCP type 0xc would be directed through slave 1, type 0x8 through slave 2, type 0x4 through slave3, and type 0x0 through slave4. Using MGEN, Scooter sent four flows of UDP traffic to Skinny. The flows were assigned DSCP values 0x0, 0x4, 0x8, and 0xc.

Router	Interface	DSCP 0x0	DSCP 0x4	DSCP 0x8	DSCP 0xc
Datagram	1	1	10	1	10
	2	1	1	10	1
Stanley	0	1	10	1	10
	1	1	10	10	10
	2	10	1	10	10
	3	10	1	1	10
Surplus1	0	1	1	10	1
	1	10	10	1	10
Manager	0	10	1	1	10
Slave1	0	1	10	10	10
	1	1	10	10	10
Slave2	0	10	1	10	10
	1	10	1	10	10
Slave3	0	10	10	1	10
	1	10	10	1	10
Slave 4	0	1	1	1	1
	1	10	10	10	1
Apollos	1	1	10	10	10
	2	10	1	10	10
	3	10	10	1	10
	4	10	10	10	1

Table 6-1 Link metrics assigned to routers in Figure 6-[illegible]

Expected outcome: Unicast data packets from source, scooter, to sink, skinny, should be routed via four distinct paths assigned by the DSCP codepoints

Measurements, data processing, and outcome: Network routing tables were periodically dumped and the tables matched the expected outcome. Tcpdump was run on the routers manager, surplus1, and slave 1 through 4 and the correct DSCP traffic was being forwarded on the correct routers. In addition, *Insure* was run on Datagram and Manager to test the code on a single area router and an area border router. All tests performed by *Insure* were passed

Test name: DSCP interoperability

Objectives: Confirm the correct operation of unicast DSCP routing in a edge routing environment.

Equipment used: Eight Linux routers, one Cisco router, one traffic source, and two traffic sinks are connected in the formation of Figure 6-2

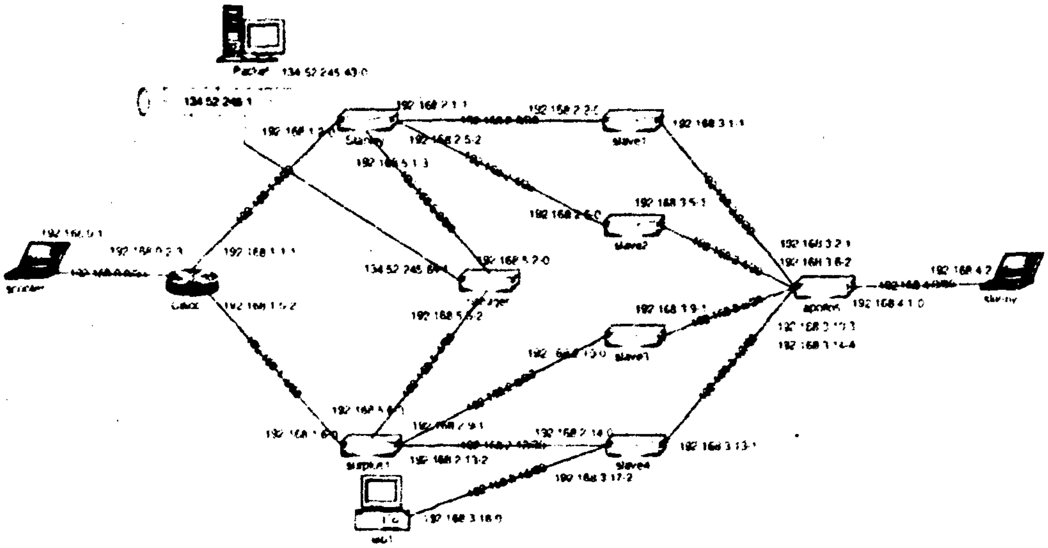


Figure 6-2 Multiple DSCP paths test configuration

Test procedures: The routers are configured with the same costs used in Figure 6-1 above except the Cisco router's interfaces can only be assigned a cost for DSCP 0x0. The Cisco router was assigned the same cost that Datagram was assigned for DSCP 0x0. Using MGEN, Skinny sent four flows of UDP traffic to Scooter and Lab1. The flows were assigned DSCP values 0x0, 0x4, 0x8, and 0xc.

Expected outcome: Unicast packets from Skinny to Scooter should be routed via a DSCP 0x0 path since the Cisco router does not support TOS based routing. Unicast DSCP enabled packets from Skinny to Lab1 will be routed according to the DSCP-enabled routes since a path of all Linux routers can be formed.

Measurements, data processing, and outcome: Network routing tables were periodically dumped and the tables matched the expected outcome. Tcpdump was run on each of the Linux routers, and the correct traffic was being forwarded on these routers. This test worked as planned.

Test name: DSCP interoperability with mixed DSCP enhancement

Objectives: Confirm the correct operation of unicast DSCP routing in a edge routing environment when the Linux routers have been enabled to use DSCP 0x0 as a DSCP value on non-DSCP routers.

Equipment used: The network configuration is identical to Figure 6-2.

Test procedures: The routers are configured with the same costs used in Figure 6-1 above except the Cisco router's interfaces can only be assigned a cost for DSCP 0x0. The Cisco router was assigned the same cost that Datagram was assigned for DSCP 0x0. Each of the Linux routers were configured in allow mixed metrics mode so DSCP routing can be used with a Cisco router. The following procedure was run when all interfaces were configured to broadcast and then repeated with all interfaces configured to point-to-point.

Expected outcome: Unicast packets from Skinny to Scooter should be routed using DSCP routes. The DSCP values for 0x4, 0x8, and 0xc on the Cisco router should appear to be set to the DSCP 0x0 value.

Measurements, data processing, and outcome: Network routing tables were periodically dumped and the tables matched the expected outcome. Tcpdump was run on each of the Linux routers, and the correct traffic was being forwarded on these routers. This test worked as planned.

Test name: Multi-area operation using DSCP

Objectives: Confirm DSCP OSPFv2 functions correctly with multiple areas when interfaces are set to broadcast, point-to-point (pp), non-broadcast multiple access (nbma), point-to-multipoint (ptmp), and point-to-multipoint multicast (ptmpmcast).

Equipment used: Nine Linux routers, a traffic source, and two traffic sinks are connected as shown in Figure 6-3.

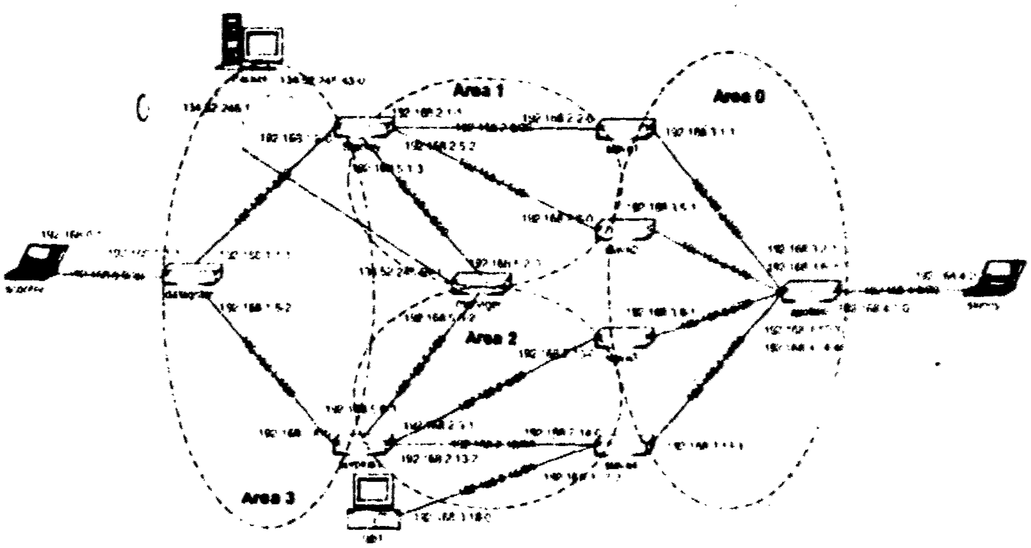


Figure 6-3 Multi-area operation using DSCP

Test procedures: The routers in Figure 6-3 were configured with the DSCP parameters shown in Table 6-1. All interfaces that are not shown were set to one for each of the four DSCP values. Eleven virtual links were configured because Manager, Stanley, and Surplus1 are non-backbone area border routers (ABR). The virtual link configuration is show in Table 6-2.

MGEN was used to generate four flows of CBR traffic from skinny to each of the sinks: Packet, Lab1, and Scooter.

This same procedure was performed on broadcast, pp, nbma, ptmp, and ptmpmcast.

ABR	Link End Points
Stanley	Slave1, Slave2, Manager, Surplus1
Manager	Slave1, Slave2, Slave3, Slave4, Stanley, Surplus1
Surplus1	Slave3, Slave4, Manager, Stanley

Table 6-2 Virtual links configured in Figure 6.2-3

Expected outcome: The summary LSAs exchanged between areas should enable DSCP-based routing to take place between the areas.

Measurements, data processing, and outcome: Network routing tables were dumped on each of the routers and compared to the expected outcome. The routing tables were correct for each of the four DSCP values for each of the 5 interface types. Tcpdump was run on the incoming interfaces of selected routers, and it was determined that the correct traffic was being forwarded on these routers. In addition, *Insure* was run on Datagram and Manager to test the code on a single area router and an area border router. All tests performed by *Insure* were passed.

Test name: Multi-area operation using DSCP with mixed metrics

Objectives: Confirm DSCP OSPFv2 functions correctly with multiple areas while some routers have partial DSCP costs defined.

Equipment used: The network configuration is shown in Figure 6-3.

Test procedures: The routers in Figure 6-3 were configured with the DSCP parameters shown in Table 6-3. All interfaces that are not shown were set to one for each of the four DSCP values. Entries in Table 6-3 that contain a "X" were not defined. Eleven virtual links were configured because Manager, Stanley, and Surplus1 are non-backbone area border routers (ABR). The virtual link configuration is show in Table 6-3.

This procedure was performed when interfaces were configured as broadcast and ptmpmcast.

Router	Interface	DSCP 0x0	DSCP 0x4	DSCP 0x8	DSCP 0xc
Datagram	1	1	10	1	10
	2	1	1	10	1
Stanley	0	1	10	1	10
	1	1	10	10	10
	2	10	1	10	10
	3	10	X	1	10
Surplus1	0	1	1	10	1
	1	10	10	1	10
Manager	0	10	X	1	10
Slave1	0	1	10	10	10
	1	1	10	10	10
Slave2	0	10	1	10	10
	1	10	X	10	10
Slave3	0	10	10	1	10
	1	10	10	1	10
Slave 4	0	1	1	1	1
	1	10	10	10	1
Apollos	1	1	10	10	10
	2	10	X	10	10
	3	10	10	1	10
	4	10	10	10	1

Table 6.2-3 Link metrics assigned to routers in Figure 6.2-3

Expected outcome: Routing table entries for DSCP 0, 8, and 12 should remain unchanged. Any DSCP 0x4 route that would have used link 192.168.5.0 or 192.168.3.4 should be rerouted and a DSCP 0x4 route to 192.168.5.0 and 192.168.3.4 should not be created.

Measurements, data processing, and outcome: Network routing tables were dumped on select routers and compared to the expected outcome. The routing tables were correct for each of the four DSCP values for broadcast and ptmpmcast.

Test name: DSCP-based multicast routing

Objectives: Confirm operation of MOSPF with DSCP.

Equipment used: The configuration shown in Figure 6-1 was used to test multicast routing in a single area configuration.

Test procedures: DSCP metrics were assigned to the links in Figure 6-1 per Table 6-1. All interfaces not defined in the table were assigned a value of one for each DSCP. Only broadcast interfaces were tested in this configuration.

Multicast traffic was originated from Skinny using MGEN. Multicast receivers, Packet, Lab1, and Scooter, responded to IGMP queries, leading to the Linux routers originating group-membership-LSAs. Multicast traffic was first originated with no DSCP markings, and the routes taken were observed. Next, the multicast caches were flushed from all the routers and the tests were repeated with Skinny originating DSCP 0x4. Finally, this procedure was repeated for DSCPs 0x8 and 0xc.

Expected outcome: The presence of DSCP-based metrics should enable the multicast tree to be built for a particular DSCP value. Figure 6-5 shows the multicast trees that should be built for each of the four DSCP metrics.

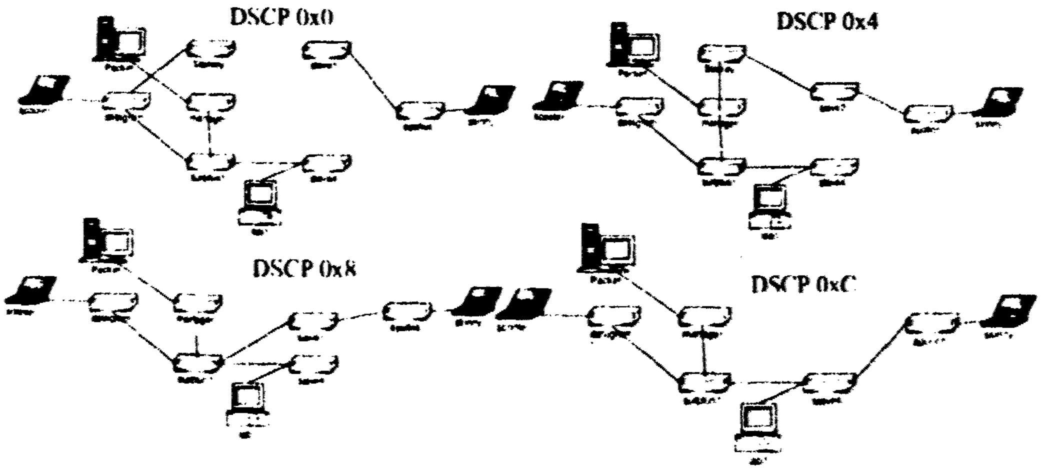


Figure 6-5 Multicast routing tree for the fourDSCP-based metrics

Measurements, data processing, and outcome: Multicast routing cache entries were dumped at each of the routers, and the routes matched the expected outcome and the multicast tree was correct. Tcpdump was run on the incoming interfaces of selected routers, and the correct traffic was being forwarded on these routers. Also, each of the multicast sinks received the multicast data.

Test name: DSCP-based multicast routing with mixed metrics

Objectives: Confirm DSCP MOSPF functions correctly while some routers have partial DSCP costs defined

Equipment used: The configuration shown in Figure 6-1 was used to test mixed metric multicast routing in a single area configuration

Test procedures: The routers in Figure 6-3 were configured with the DSCP parameters shown in Table 6-4. All interfaces that are not shown were set to one for each of the four DSCP values. Entries in Table 6-4 that contain a "X" were not defined. Eleven virtual links were configured because Manager, Stanley, and Surplus1 are non-backbone area border routers (ABR). The virtual link configuration is show in Table 6-3. This procedure was performed when interfaces were configured as broadcast

Router	Interface	DSCP 0x0	DSCP 0x4	DSCP 0x8	DSCP 0xc
Datagram	1	1	10	1	10
	2	1	1	10	1
Stanley	0	1	10	1	10
	1	1	10	10	10
	2	10	1	10	10
	3	10	1	1	10
Surplus1	0	1	1	10	1
	1	10	10	1	10
	2	1	1	1	X
Manager	0	10	1	1	10
	2	1	1	1	X
Slave1	0	1	10	10	10
	1	1	10	10	10
Slave2	0	10	1	10	10
	1	10	1	10	10
Slave3	0	10	10	1	10
	1	10	10	1	10
Slave 4	0	1	1	1	1
	1	10	10	10	1
Apollos	1	1	10	10	10
	2	10	1	10	10
	3	10	10	1	10
	4	10	10	10	1

Table 6 4 Link metrics assigned to routers in Figure 6.2-3

pected outcome: Routing table entries for DSCP 0, 8, and 12 should remain unchanged. Any DSCP 0xc route that would have used link 192.168.5.4 should be rerouted and a DSCP 0xc route to 192.168.5.4 should not be created

Measurements, data processing, and outcome: Network routing tables were dumped on select routers and compared to the expected outcome. The routing tables were correct for each of the four DSCP values.

Test name: Multi-area multicast operation using DSCP

Objectives: Confirm DSCP MOSPF functions correctly with multiple areas

Equipment used: Figure 6-3 displays the equipment configuration, with multicast source, Skinny, and group members, Packet, Lab1, and Scooter.

Test procedures: The network was configured identically to the Multi-area operation using DSCP test

Expected outcome: Despite the fact that route summarization occurs across area boundaries, multicast routing should take into account the DSCP metrics in computation of shortest-path trees. Figure 6-5 shows the multicast trees that should be built for each of the four DSCP metrics

Measurements, data processing, and outcome: Multicast routing cache entries were dumped at each of the routers, and the routes matched the expected outcome and the multicast tree was correct. Tcpdump was run on the incoming interfaces of selected routers, and the correct traffic was being forwarded on these routers. Also, each of the multicast sinks received the multicast data. The tests were passed for each of the 5 interface types.

Test name: Wireless and DSCP OSPFv2

Objectives: Confirm interoperability of Wireless OSPFv2 and DSCP routing (both Boeing modifications operating simultaneously)

Equipment used: We will reuse the configuration shown in Figure 6-2 above

Test procedures: The network will configured as in Figure 6-2 and Wireless OSPFv2 will be used as the routing protocol. Two flows of CBR traffic will be sent from User 1 to User 2. One flow will be marked with DSCP type 2, and it will be directed across the low bandwidth link. The other flow with DSCP 4 will be directed to the wireless network Network connectivity will be varied to verify correct routing. Similar tests will be conducted with multicast data and multicast routing

Measurements and data processing: Network routing tables will be periodically dumped and post-processed, and compared against the expected behavior. Tcpdump will be run on each interface and saved to a log file, which will be post-processed to extract the delivery ratio and end-to-end delay of the user traffic based on the DSCP type. In addition, the amount of overhead generated by Wireless OSPFv2 will be calculated.

Expected outcome: Traffic with DSCP type 2 should take the low bandwidth link when it is in the up state and transition to the wireless network when it is in the down state. The network should be able to be able to obtain a higher delivery ratio than obtained in the Heterogeneous Network when using only Wireless OSPFv2.

Test name: DSCP and QoS scheduling integration

Objectives: Confirm interoperability between DSCP-based routing and built-in Linux DSCP-based packet (queue) schedulers.

Equipment used: The test configuration of Figure 6-1 will be reused

Test procedures: User 1 sends four flows of CBR traffic to user 2. Each flow is assigned a different DSCP value Link metrics shall be arranged such that the different flows find different paths. In addition, certain DSCP codepoints will receive preferential treatment in priority queues.

Measurements and data processing: Tcpdump will be run on each interface and saved to a log file, which will be post-processed to extract utilization statistics, and queuing priority. Network routing tables will be periodically dumped and post-processed, and compared against the network emulation scenario.

Expected outcome: Data packets from users 1 to 2 will be routed via different paths according to their DSCP codepoint In addition, we should observe flows with different DSCP codepoints receiving different service in the queues

6.2 Summary

This section has described our testing results for the DSCP-based dynamic routing We observed the following characteristics when testing the implementation.

- The Linux kernels used in testing were patched to support all DSCP codepoint values, as described in Section 4.3.1. The stock kernels that ship with the versions of RedHat Linux that we used, along with the latest kernel available from kernel.org, still use TOS values for routing, limiting routing entries to eight possible TOS values Patched kernels allow all 64 possible DSCP values to be used for routing entries.
- The routing support in the kernel for multicast traffic does not allow multiple multicast routing entries for the same source and multicast group pair. This is a kernel limitation, not an implementation limitation, and we did not experiment with kernel extensions to support multiple entries per source and multicast group pair

This implementation leveraged legacy "TOS" fields in the OSPFv2 LSAs OSPFv3 for IPv6 does not support these fields, so DSCP-based metric information must be disseminated in additional LSAs (such as opaque LSA types) Furthermore, it appears that multicast protocols must be extended to use this capability if MOSPF is not the multicast protocol in use (specifically, PIM and IGMP may need modifications to support different routing based on the DSCP value in the data packets). Therefore, while this implementation is a proof-of-concept implementation, further protocol development work is needed for this capability to be implemented in IPv6 or multicast protocols other than MOSPF

7 Demonstration results

7.1 Scope and objectives of demonstration

Our demonstration attempted to meet both *broad program objectives* and *specific technical objectives*, in demonstration scenarios that map to NBN operational concepts, as described below.

7.1.1 Program objectives

The Boeing Composite Routing program does not focus on technology development that covers all aspects of the broad operational goals identified in the program solicitation, but our demonstration program was designed to leverage off-the-shelf components in the areas not specifically addressed by the routing protocol implementation. Table 7-1 briefly describes how NBN Composite Routing overall program objectives were met by a combination of off-the-shelf hardware and software components and the new routing protocol implementation, and provides pointers to later subsections. Sub-demos are described in Section 7.3.

NBN Composite Routing goal	Demonstration concept	Relevant sub-demo
(i) Any onboard IP data routed to any communication link (including Link 16) based on the latency requirement	• **wireless OSPF interface type enables bandwidth-efficient operation on multihop wireless networks**	1, 2, 4, 5
	• **DSCP-based routing allows for dynamic policy routing for load balancing and QoS control**	2, 3
	• Note: IP over Link 16 not addressed by this program	not applicable
(ii) automated network participation (i.e., leaving and joining a network with no manual operations)	• mobile wireless nodes join network via DHCP	4, 5
	• DHCP authentication via RFC 3118	4, 5
(iii) all traffic authenticated using either network or application layer security	• show compatibility of protocols with TACLANE surrogates (IPsec gateways)	2
	• all router messages authenticated using MD5 authentication	2, 3, 4, 5
	• host authentication and encryption via Host Identity Protocol	5
(iv) automated network management and dynamic bandwidth management	• router contains SNMP-compatible MIB and user agent	2, 3
	• SNMP element manager for device monitoring	2, 3
	• Note: Dynamic bandwidth management (above and beyond dynamic routing) not addressed by this program	not applicable

Table 7-1 Mapping of program goals to demonstration concepts. ***Bold font*** *indicates main program software deliverables*

7.1.2 Technical objectives

The main technical objectives of the demonstration include:

(i) demonstrate the correctness of the new routing protocol implementation by dynamically changing the network topology;

(ii) compare the performance of the legacy ADNS routing protocol (OSPFv2) against the new implementation; and

(iii) illustrate how routers based on our extensions can interoperate with legacy equipment.

Our specific technical objectives are shown in Table 7-2

Technical demonstration goal	Sub-goals	Relevant sub-demo
Wireless OSPF dramatically reduces routing protocol overhead in wireless networks	• N/A	1
Show operation of wireless OSPF extensions in heterogeneous network environment	• support for various link technologies (satellite, LOS, ad hoc, point-to-point)	2
	• mixed node (wireless and wired interfaces) operation	2
	• multicast compatibility with wireless OSPF	3
	• host and platform mobility compatible with OSPF routing	4, 5
DSCP-based dynamic routing can improve Navy load balancing	• show load-balancing of unicast traffic, including mixed operation with BGP routers	2
	• show multicast operation (MOSPF) for notional Link-16 range extension	3
Interoperability	• show interoperability with Cisco BGP routers	2
	• show wireless and DSCP extensions working simultaneously	2

Table 7-2 Summary of technical demonstration goals

7.1.3 Metrics

The demonstration and experiments were instrumented to provide real-time or post-processed statistics on the network performance. Much of the demonstration was focused on displaying a new capability (such as load balancing) rather than an improvement on an existing capability, so we did not have a quantitative metric associated with them. Table 7-3 lists the quantitative metrics that were measured during portions of the demonstrations described in Section 5.

Metric	Definition
Control overhead	Number of control bits or packets (relating to routing protocol operation) transmitted during the demonstration run.
Packet delivery ratio	Ratio of user data packets sent into the network to data packets received by the destination nodes
Routing convergence time	Time between establishment of link layer connectivity and IP addressing and the time at which IP routing is stable enough for data transfer to occur

Table 7-3 List of metrics measured in demonstrations and experiments

7.2 Integrated Testbed Configuration

Most of the demonstrations described in Section 7.3 made use of the integrated testbed shown in Figure 7-2, designed to show the following capabilities:

- operation of wireless interface type on actual wireless networks (support for various link technologies, mixed node operation, and multicast operation)
- operation of DSCP-based routing (including an operationally relevant DSCP marking strategy, load-balancing of unicast traffic compatible with BGP routers, and multicast operation)
- automated network participation (leave and join)
- traffic authentication and security
- automated network management
- interoperability with legacy routers, and between the wireless and DSCP extensions

Figure 7-1 provides a notional representation of the demonstration. The demonstration testbed highlighted IP networking between ships, over satellite links and via ship-to-ship wireless networking. Specifically, the demonstration highlighted the notional concepts of preferential dynamic routing of JCA traffic over Challenge Athena (in the presence of in-line encryptors), range extension of Link-16 over the ADNS system, and multihop wireless line-of-sight/beyond-line-of-sight (LOS/BLOS) networks supporting host and platform (ship) mobility.

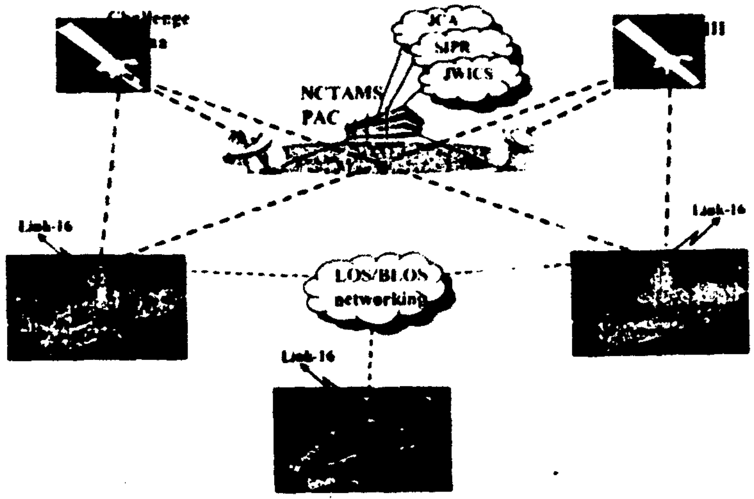


Figure 7-1 Integrated testbed notional concept

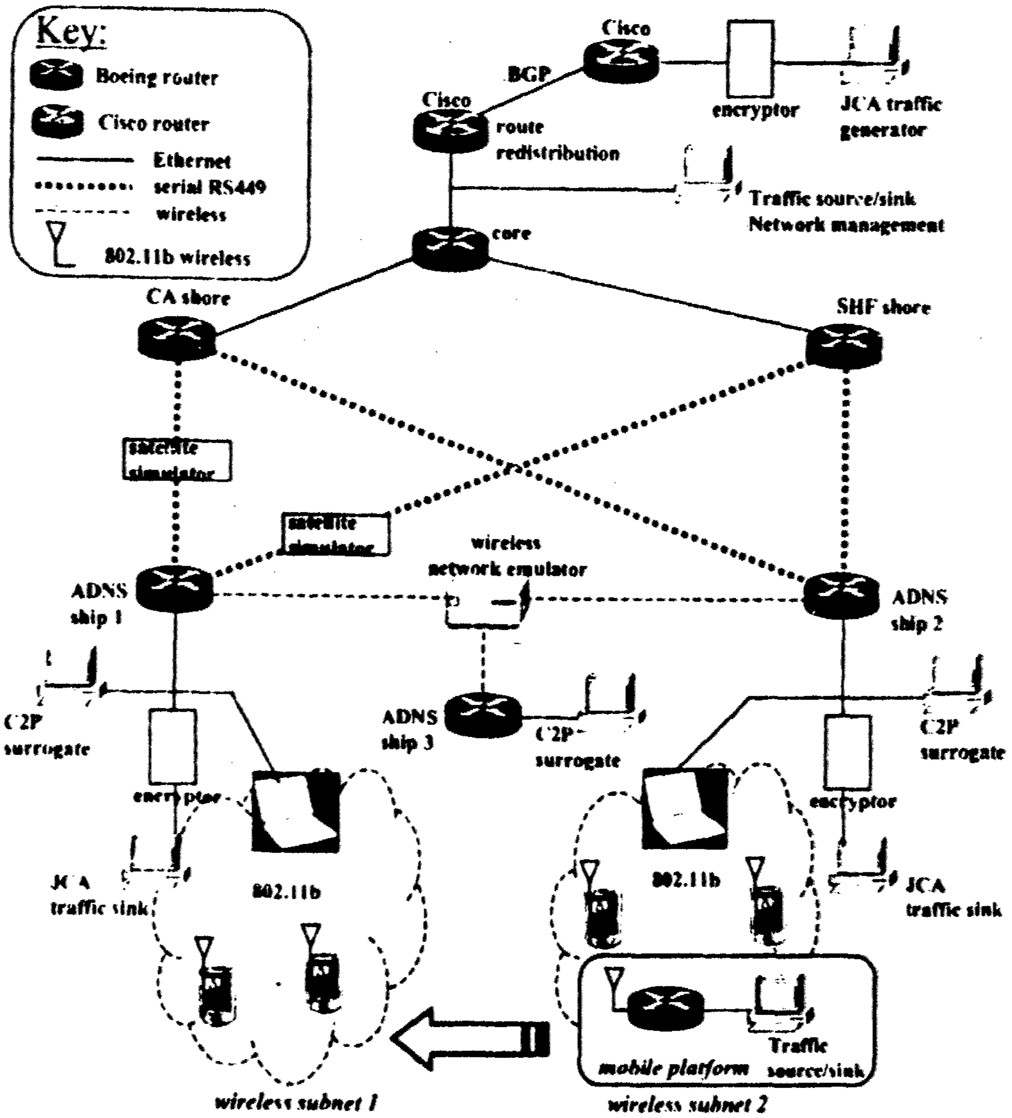


Figure 7-2 Integrated testbed equipment

Figure 7-2 illustrates how the notional concept in Figure 7-1 was instantiated. The following equipment was part of the demonstration testbed

- **Linux router.** Linux routers contain one or more wired or wireless interfaces, and a modified OSPFv2 routing daemon. Wired interfaces were either Ethernet or serial RS-449 (point-to-point). Wireless interfaces were either 802.11b or an Ethernet interface plugged into the wireless network emulator. A variety of computers were used as routers, including Dell PowerEdge rackmount servers, Dell Optiplex desktops, custom-built PC's, laptops, and iPAQ handheld computers.
- **Cisco router.** We used two Cisco Mobile Access Router (MAR) routers running OSPFv2 and RIPv2.
- **Satellite simulator.** We used two Adtech SX/13 hardware link emulators with RS-449 interfaces
- **Traffic sources/sinks.** These were Linux-based workstations running traffic generation utilities such as NRL's MGEN and IP-based video cameras or display programs. JCA traffic was represented by a long-running TCP transfer
- **C2P surrogate.** The enhanced C2P in the future will permit Link-16 tracks to be carried over ADNS's IP infrastructure, using multicast routing. For demonstration purposes, we did not send sensor tracks but

- instead used multicast video from a single source (ship 1) to multiple receivers (ships 2 and 3) as a surrogate traffic type.
- **Encryptor.** Our original plan to use TACLANE encryptors was no longer feasible since the demonstration location changed to San Diego. However, we constructed IPsec-based gateways that are functionally equivalent to TACLANEs, using Linux IPsec. TACLANE Release 1.1 supports IP Type-of-Service (TOS) bits bypass, and the DSCP field is a redefinition of the TOS field.
- **Wireless network emulator.** We used the Boeing Synthetic Network Environment (SNE) as the wireless emulator. The SNE runs a mobile ad hoc emulation script that emulates the connectivity of mobile nodes. When two nodes are deemed to be out of radio range from one another in the emulation, the SNE blocks packet reception from one another respectively. The SNE is based on HRL's mobiemu tool,[2] and has been extended to also support bandwidth limitation and packet errors or (deterministic or statistical) losses. The mobility scripts are created through the use of CMU's scenario generation tool (part of the ns-2 Internet simulator) and can be run at different speeds to effect different mobility.

7.3 Demonstration conduct

We conducted final program demonstration on September 18, 2003, in Building 91 at SPAWAR SSC-SD. The demonstration consisted of five "sub-demos." The first was a focused demonstration of wireless routing (Section 7.3.1), followed by several demonstrations using the integrated routing testbed (Sections 7.3.2-7.3.5) showing both wireless routing and DSCP-based routing in the context of NBN operations.

7.3.1 Wireless routing demonstration (Sub-demo 1)

This demonstration was a focused comparison of the performance of the new wireless interface type for OSPFv2 with that of the legacy Point-to-Multipoint interface type.

7.3.1.1 Objective and expected results

Operation of legacy OSPFv2 over multihop wireless networks requires configuration of the interface into Point-to-Multipoint mode. This mode of operation leads to high overhead because of the requirement to maintain an adjacency pair-wise between nodes.

The wireless interface type reduces the amount of overhead used to distribute routing information. In Point-to-Multipoint mode (the only other OSPFv2 configuration appropriate for a wireless multihop subnet), each router must form an adjacency with every other router. Because every router must form a full adjacency and synchronize its databases with every other router in Point-to-Multipoint mode, the overhead grows at an exponential rate as more nodes are added to the network. Furthermore, traditional OSPF uses a reliable flooding algorithm to distribute link state advertisements (LSAs) through the network. In contrast, the wireless interface type does not require full adjacencies between routers, and uses an optimized, unreliable flooding algorithm for efficient LSA distribution.

When using the wireless interface, we expected up to an order of magnitude reduction in the amount of overhead generated (when compared to the Point-to-Multipoint mode), without a significant decrease in the packet delivery ratio performance.

7.3.1.2 Experiment equipment and configuration

The experiment requires the use of 16 routers and a network emulator. Rather than use 16 physical routers, we used 16 virtual routers, each hosted on a physical router using the VMware product for virtual machine hosting. Specifically, four physical machines hosted 16 virtual routing processes, as if there were four physical machines present for each actual machine. The OSPF configuration was that of a single subnet (single area, single autonomous system).

The network emulator (Boeing Synthetic Network Environment--SNE-- described in Section 4) provided an emulated multihop radio environment based on a preconfigured mobility script. We used a centralized version of the emulator, supported by 16 Fast Ethernet ports on a Linux-based bridge. The mobility scripts were designed to

[2] Zhang, Y. and W. Li, "An Integrated Environment for Testing Mobile Ad Hoc Networks." *Proceedings of ACM MobiHoc Conference*, 2002. (Available on-line at http://www.wins.hrl.com/people/ygz/papers/mobihoc02b.html)

suitably stress the routing protocol. Although the routers were connected via Ethernet to the wireless emulator, the router interfaces were configured as either Point-to-Multipoint or wireless OSPF.

To test the ability of the routers to deliver user traffic, each node generated a UDP packet and sent it to each other node in the emulation, once per second. Assuming this is a minimal 10 byte packet, this resulted in a rough maximum of 16*15*(40 + 10)*8 = 100 Kb/s of traffic generated. The bandwidth of the emulated wireless channel was not artificially restricted by the emulator.

Figure 7-3 illustrates the equipment configuration. In addition to the Linux routers and the emulator, we used an additional display to provide visualization of the demonstration

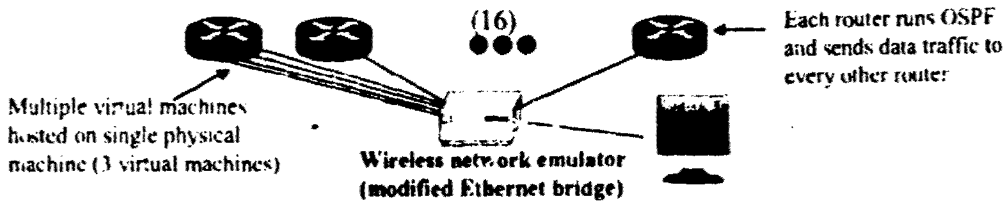


Figure 7-3. Demonstration configuration for wireless network emulator

7.3.1.3 Demonstration procedures

Prior to the demonstration, we generated 120-second mobility scenarios for the test configuration. The first demonstration consisted of operation in legacy OSPF (Point-to-Multipoint mode). The OSPF daemons were first started on each router. Next, the emulation script was executed. Following completion of the emulation, data was gathered from each router, processed, and the results visually displayed. After finishing of the Point-to-Multipoint configuration, the OSPF daemons were restarted in wireless OSPF mode, and the above demonstration was re-run.

7.3.1.4 Metrics and data collection

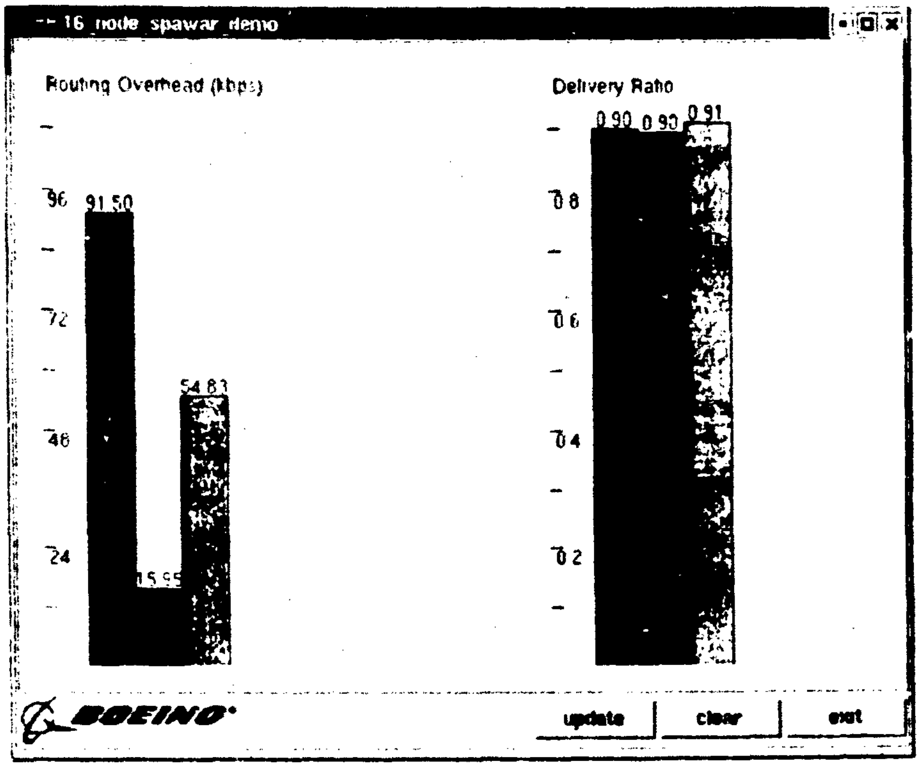
The two metrics that were displayed were <u>routing protocol overhead</u> and <u>packet delivery ratio</u>.

<u>Routing protocol overhead</u> counts all OSPF packets sent into the channel, including a breakdown between unicast and multicast link layer frames. The overhead was determined by counting the total number of bytes and packets sent during the emulation, divided by the number of seconds of emulation. The number of packets sent was counted on each node using tcpdump and shell and Perl scripts. Specifically, during the conduct of the demonstration, each node performed a tcpdump on packets sent on its wireless interface. Upon the end of emulation, the tcpdump terminated and scripts automatically ran to post-process the packet dumps and extract the desired statistics. These statistics were queried for from the master emulator, which displayed them in a graphical format.

<u>Packet delivery ratio</u> was defined as the total number of packets successfully received by the routers, divided by the number sent. Packets sent into the network were either be delivered successfully, or were dropped due to a lack of route to the destination, or were dropped due to a loss on the link. Each router kept count of the number of packets that it received from each of its peers. At the end of the emulation, the master emulator queried each node for the packet reception counts, and displayed the results in a graphical format.

7.3.1.5 Demonstration results

As expected, we observed that the overhead performance of wireless OSPF was between that of OLSR and OSPF with Point-to-Multipoint interfaces. In our 16-node demo, we observed roughly 16 Kb/s of overhead for OLSR, 55 Kb/s of overhead for wireless OSPF, and 92 Kb/s for OSPF Point-to-Multipoint. The packet delivery was slightly higher for wireless OSPF-- also expected. During the actual demonstration, one unexpected result was that the wireless OSPF overhead was higher than what we had observed the previous day, and higher than what we had observed in simulations. After the demonstration, we tracked this discrepancy to a bug in the implementation and fixed it. Section 5-2 above shows the results of our final implementation testing.


16 node spawar demo
Routing Overhead (kbps)
96
72
48
24
91.50
15.95
54.83
Delivery Ratio
0.8
0.6
0.4
0.2
0.90
0.90
0.91
BOEING
update
clear
exit

7.3.2 DSCP and wireless routing integration (Sub-demo 2)

This demonstration was designed to show the operational benefit of DSCP-capable dynamic routing, and to show the integration of the DSCP and wireless extensions to OSPF.

7.3.2.1 Objective and expected results

We expect to show the following results with this demonstration:

- DSCP-based dynamic routing allows for load balancing across satellite links with asymmetric bandwidth, in a manner that allows for dynamic failover to alternate paths should the primary path for a class of traffic fail;
- DSCP-based routing is interoperable with DSCP-based packet scheduling on congested links; and
- DSCP-based routing and wireless OSPF extensions are interoperable.

We will repeat our demonstration conducted at SPAWAR in May, 2003, with three extensions. The demonstration in May showed how JCA traffic could be priority routed over notional Challenge Athena links, and that fallback paths were in place to route (and priority schedule) the JCA traffic over line-of-sight wireless links if the satellite connectivity to a particular ship completely failed. The three extensions to this demonstration are as follows:

(i) we will illustrate successful operation of this capability through in-line packet encryptors;

(ii) we will show successful "mixed-mode" operation using AS-external LSAs (a feature that was not implemented in May, 2003); and

(iii) we will replace the point-to-point link between ships with an emulated wireless multihop network.

7.3.2.2 Experiment equipment and configuration

We will use the subset of the integrated demonstration configuration (Figure 7-2) shown in Figure 7-4. As Figure 7-4 illustrates, the demonstration will require six Boeing routers (additional wireless routers may be added via virtual machines), two Cisco routers, three encryptors, and six traffic sources and sinks. The links will be composed of Ethernet segments, two satellite simulators, six serial RS449 connections, and a wireless network emulator.

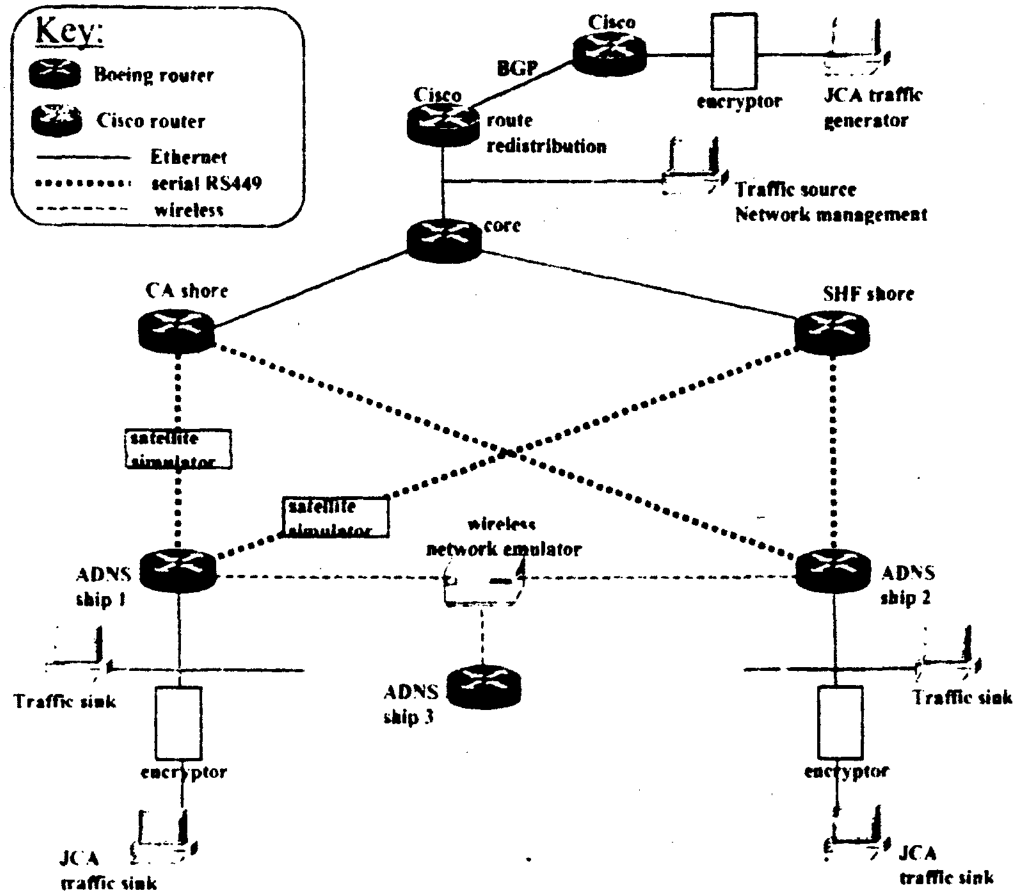


Figure 7-4 Demonstration configuration for "DSCP and Wireless Routing Integration."

The machines will be configured as follows:

- **JCA traffic sources and sinks.** These Linux machines will generate long-running TCP connections flowing from the JCA source to the JCA sinks on each ship. The TCP connections will be marked with a selected DSCP value.
- **Traffic source and sinks.** These Linux machines will generate and consume test traffic with DSCP values different from the value used for JCA traffic.
- **Network management.** This Linux machine will also serve as a network management workstation with a graphical depiction of the network topology.
- **Encryptor.** These Linux machines will perform packet encryption similar to TACLANE encryptors.
- **Wireless network emulator.** This machine will provide emulation of a multihop radio environment.
- **Satellite simulator.** These machines will provide delay and bandwidth emulation between routers.
- **Cisco routers.** These routers will connect the JCA traffic source to the rest of the network using BGP and route redistribution into OSPF.
- **Linux routers.** These routers will be the focus of the demonstration. Each will run an instance of OSPF routing with DSCP-based and wireless extensions. The DSCP extensions will permit the setting of link metrics on a per-DSCP basis, thereby influencing the packet flows for different DSCP classes. The wireless extensions will operate efficiently over the wireless multihop network. In addition, standard Linux traffic controls will permit priority queuing techniques on the basis of DSCP.

7.3.2.3 Demonstration procedures

All of the Linux-based OSPF processes will be configured to operate in "mixed-mode" operation, which means that they will be able to compute DSCP-enabled routes despite the presence of non DSCP-capable routers (Cisco) in the path. The following steps will be taken to set up the demonstration:

(i) configure the packet encryptors for encrypted tunnels (JCA source to JCA sinks), and configure the encryptors to pass the DSCP value through to the tunneled outer IP header;

(ii) configure traffic sources to generate flows with desired DSCP values;

(iii) configure link metrics on Linux-based routers to preferentially route emulated JCA traffic over Challenge Athena links, and other traffic over SHF links, with fallback routes via ship-to-ship links;

(iv) configure priority queuing on satellite interfaces such that JCA traffic is served at higher priority than other traffic when the queue is congested; and

(v) configure the wireless network emulator to provide a multihop wireless environment.

The demonstration will first illustrate the correct routing (load balancing) of JCA traffic and other traffic. Next, the Challenge Athena satellite link will be brought down to ship 1, and JCA traffic will be rerouted to the SHF link to ship 1. Next, the SHF link to ship 1 will be brought down, forcing the JCA traffic to be rerouted to ship 2, and via the wireless network to ship 1. The satellite bandwidth to ship 2 at this point will be oversubscribed, with the result that JCA traffic will receive higher priority over the congested links. The satellite links will be restored in reverse order, with the resulting routing recovering to the original state.

7.3.2.4 Metrics and data collection

The testbed will be instrumented with packet sniffing processes (tcpdump) on key interfaces in the topology. Using these tools, in combination with a graphical utility for displaying DSCP-based traffic flows on an interface (shown in Figure 7-5), we can display the operation of the protocol in real-time, and will verify the correct routing by inspection of packet traces.

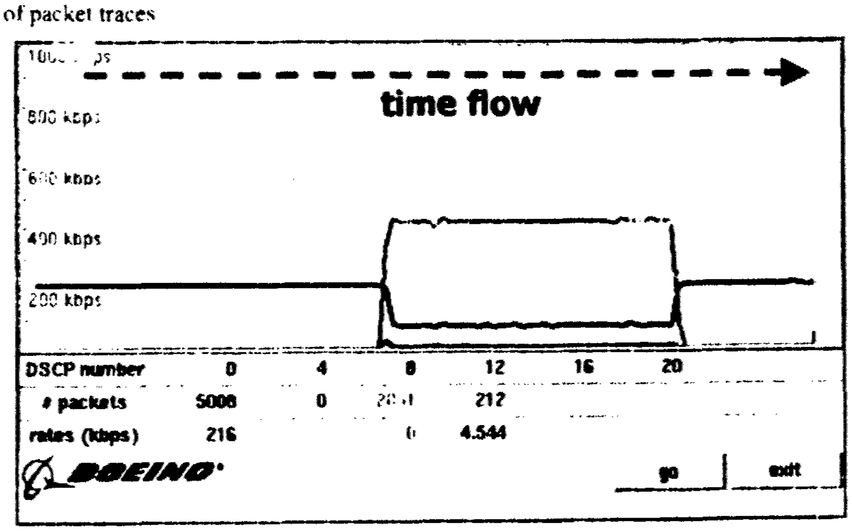


Figure 7-5 Visualization of traffic flow on a link, based on DSCP value.

7.3.3 Link-16 range extension (Sub-demo 3)

This demonstration is designed to show the operational benefit of multicast DSCP-capable dynamic routing, in the context of future use by enhanced Command and Control Processors (C2P) used for range extension of Link-16 networks over ADNS.

7.3.3.1 Objective and expected results

The NBN "Battleforce Composite Networking" program is building a Link-16 range extension capability, to enable Link-16 tracks to be sent to other enhanced Command and Control Processors (C2P) over the ADNS system. Since the prototype C2Ps with IP interfaces are not yet ready, we will use surrogate C2Ps (Linux computers with a multicast application) for this demonstration.

A key component of this system will be the use of QoS capabilities in Cisco routers to handle this high priority traffic flow. In our demonstration, we will show how DSCP-based multicast routing can help to preferentially route traffic along certain network paths.

7.3.3.2 Experiment equipment and configuration

We will use the subset of the integrated demonstration configuration (Figure 7-2) shown in Figure 7-6. As Figure 7-6 illustrates, the demonstration will require six Boeing routers (additional wireless routers may be added via virtual machines), and four traffic sources and sinks. The links will be composed of Ethernet segments, two satellite simulators, six serial RS449 connections, and a wireless network emulator. The emulated C2P machines are connected to ADNS routers at the (notional) Secret level.

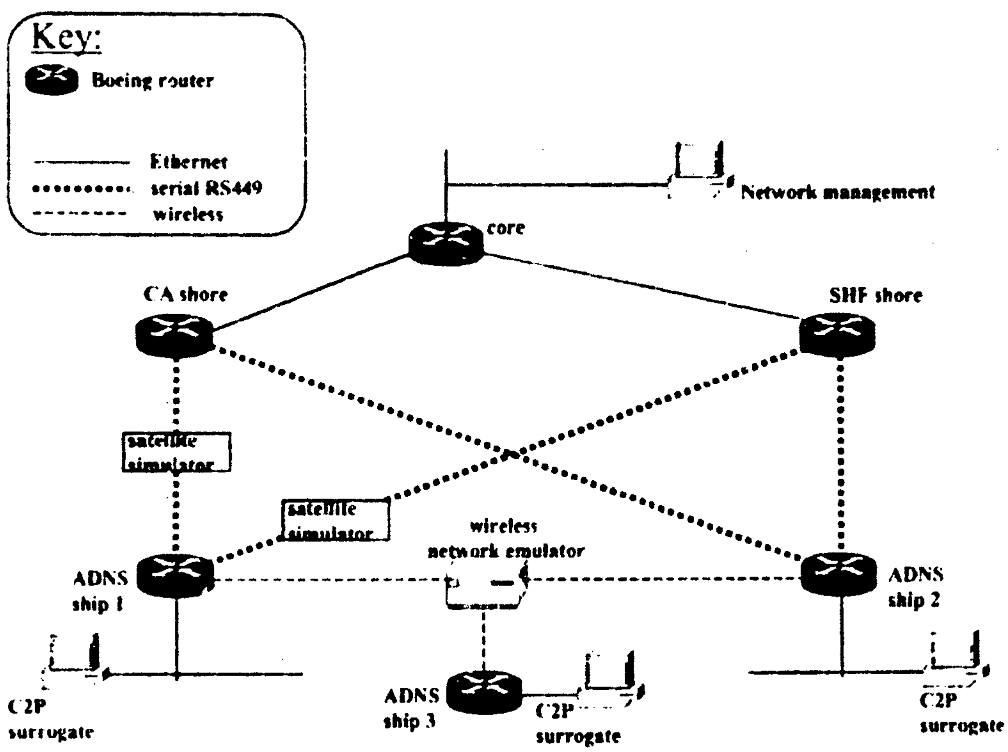


Figure 7-6 Demonstration configuration for "Link 16 Range Extension."

The machines will be configured as follows:

- **C2P surrogate.** These are Linux machines running video software (video serves as a surrogate for Link-16 tracks). The C2P surrogate on ship 2 will originate multicast video datagrams, marked with a specific DSCP value, that will be delivered to the receivers on ship 1 and ship 3.
- **Network management.** This Linux machine will also serve as a network management workstation with a graphical depiction of the network topology.
- **Wireless network emulator.** This machine will provide emulation of a multihop radio environment.
- **Satellite simulator.** These machines will provide delay and bandwidth emulation between routers.
- **Linux routers.** These routers will be the focus of the demonstration. Each will run an instance of OSPF routing with DSCP-based and wireless extensions. The DSCP extensions will permit the setting of link metrics on a per-DSCP basis, thereby influencing the packet flows for different DSCP classes. The wireless extensions will operate efficiently over the wireless multihop network. In addition, standard Linux traffic controls will permit priority queuing techniques on the basis of DSCP.

7.3.3.3 Demonstration procedures

The following steps will be taken to set up the demonstration:

(i) configure the C2P surrogate application to generate flows with the desired DSCP value;

(ii) configure link metrics on Linux-based routers to preferentially route emulated C2P traffic first over wireless links, then second over Challenge Athena links, while other traffic off ship uses SHF links;

(iii) configure the wireless network emulator to provide a multihop wireless environment.

The demonstration will first illustrate the multicast traffic using the multihop wireless network for video distribution. Next, the link from Ship 1 to the rest of the wireless network will be broken. At this point, multicast frames will additionally flow over Challenge Athena links to Ship 1.

7.3.3.4 Metrics and data collection

The testbed will be instrumented with packet sniffing processes (tcpdump) on key interfaces in the topology, and DSCP traffic monitors such as shown above in Figure 5-3. This demonstration is more a demonstration of capability rather than performance metrics so no specific performance metrics will be collected for this experiment.

7.3.4 Mobile platform (Sub-demo 4)

This demonstration is designed to show the operational goal of mobile platforms being supported by wireless OSPF routing.

7.3.4.1 Objective and expected results

This demonstration will show a mobile router with an associated subnet roaming from one network to another. A host on the mobile platform shall communicate with a fixed node in the infrastructure. The router detects when it enters the new subnet, configures its new IP address (obtained in an authenticated fashion by DHCP), restarts or reconfigures the OSPF daemon, starts advertising the attached subnet in its new wireless subnet, and data transfer starts flowing again between the host on the mobile platform and the host in the fixed infrastructure. The purpose is · w how wireless OSPF can support transit operation of mobile platforms.

7.3.4.2 . Experiment equipment and configuration

We will use the subset of the integrated demonstration configuration (Figure 7-2) shown in Figure 7-7. As Figure 7-7 illustrates, the demonstration will require five Boeing routers in the fixed infrastructure, and a number of physically mobile routers. One mobile router will be hosting a subnet with a Linux traffic source and sink, which can talk to hosts on each of the notional ships. Other mobile routers will include PDAs and laptops with 802.11b interfaces. The links will be composed of two 802.11b subnets, Ethernet segments, and a wireless network emulator.

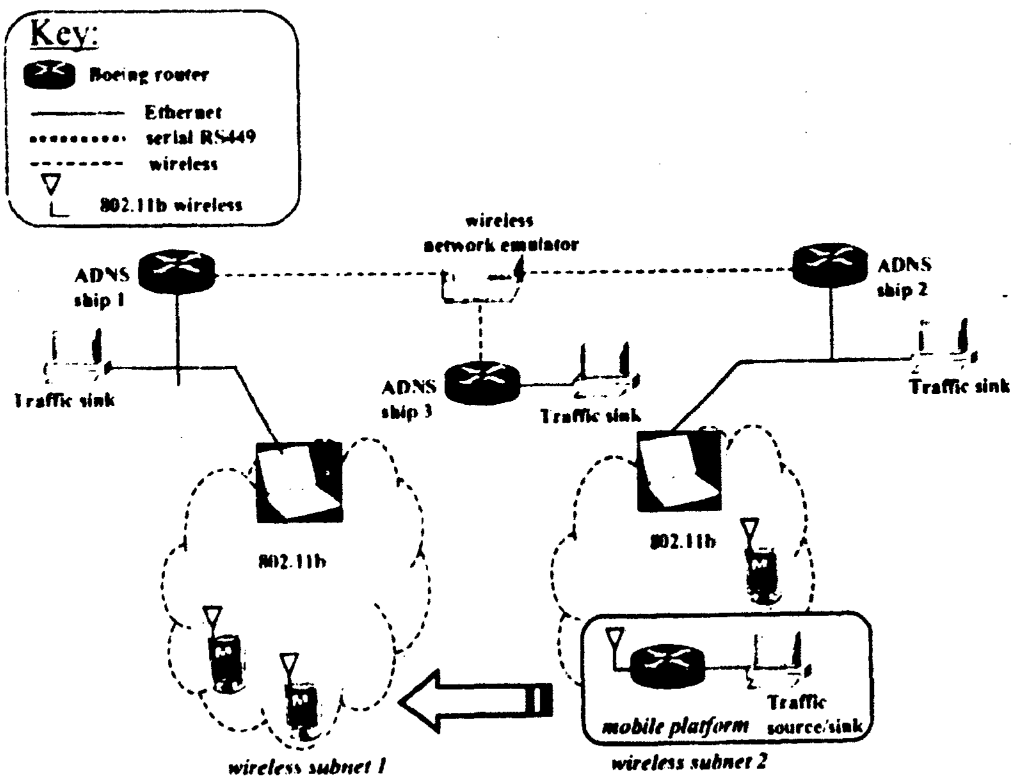


Figure 7-7 Demonstration configuration for "Mobile Platform"

The machines will be configured as follows:

- **Traffic sources and sinks.** These are Linux machines running traffic generation programs such as NRL's MGEN

- **Wireless network emulator.** This machine will provide emulation of a multihop radio environment.
- **Satellite simulator.** These machines will provide delay and bandwidth emulation between routers.
- **Linux routers.** There will be a number of different Linux-based routers in the demonstration, including Dell rackmount and desktop servers, laptops, and PDAs (Compaq iPAQ), each running an instance of the wireless OSPF daemon. In wireless subnet 1, the fixed router will also serve as a DHCP server, and the mobile platform will run DHCP client software.

7.3.4.3 Demonstration procedures

To set up the demonstration, we will configure devices on wireless subnets 1 and 2, including 802.11b channels, IP addressing, and routing protocol daemons. On the mobile plaform, the wireless router will advertise reachability to the IP subnet hosting the traffic source. The wired/wireless router on subnet 1 will be configured to serve as a DHCP server.

The mobile platform will initiate a long-running data transfer with a traffic sink on ship 2. Next, the mobile platform will be taken out of range of wireless subnet 1 (either physically out of range or by reconfiguring the 802.11b channel of operation), and into range of subnet 2. The router will acquire a new IP address in an authenticated manner, using cryptographic techniques for DHCP found in RFC 3118.[3] Upon obtaining new IP configuration for the new wireless subnet, the router will begin to advertise reachability of the subnet, and the data transfer between the traffic source and sink will resume once the routing converges.

7.3.4.4 Metrics and data collection

The key metric for this demonstration will be routing convergence time, defined as the time from which the mobile router acquires a new IP address to the time at which IP routing is stable enough for the data transfer to resume. We will estimate this by conducting tcpdumps on key interfaces in the topology, which will timestamp the significant events in this demonstration.

[3] R. Droms and W. Arbaugh, "Authentication for DHCP Messages," Internet Request for Comments (RFC) 3118, June 2001, available on-line at http://www.ietf.org/rfc/rfc3118.txt

7.3.5 Host joining and authentication (Sub-demo 5)

This demonstration is designed to show the operational goal of mobile hosts being supported by wireless OSPF routing and authenticated DHCP, and how data traffic can be authenticated and encrypted on an end-to-end basis.

7.3.5.1 Objective and expected results

This demonstration will show a mobile router with an active TCP connection to a host in the fixed infrastructure, as it changes subnets, readdresses its interface, and continues to use its TCP connection with the remote host despite the readdressing. This demonstration combines the wireless OSPF routing with the Host Identity Protocol (HIP).[4] a proposal under consideration at the IETF as a new approach to host mobility, multihoming, and authentication. The purpose is to show how wireless OSPF can be used in a network context that allows for hosts to join networks in an authenticated manner, with all traffic encrypted.

7.3.5.2 Experiment equipment and configuration

This configuration (shown in Figure 5-6) is essentially the same as shown above in Figure 5-5, but with a mobile host (wireless laptop) used in place of a mobile platform. As Figure 5-6 illustrates, the demonstration will require five Boeing routers in the fixed infrastructure, and a number of physically mobile routers. One mobile router will also be a host that talks to a host on one of the ships. Other mobile routers will include PDAs and laptops with 802.11b interfaces. The links will be composed of two 802.11b subnets, Ethernet segments, and a wireless network emulator

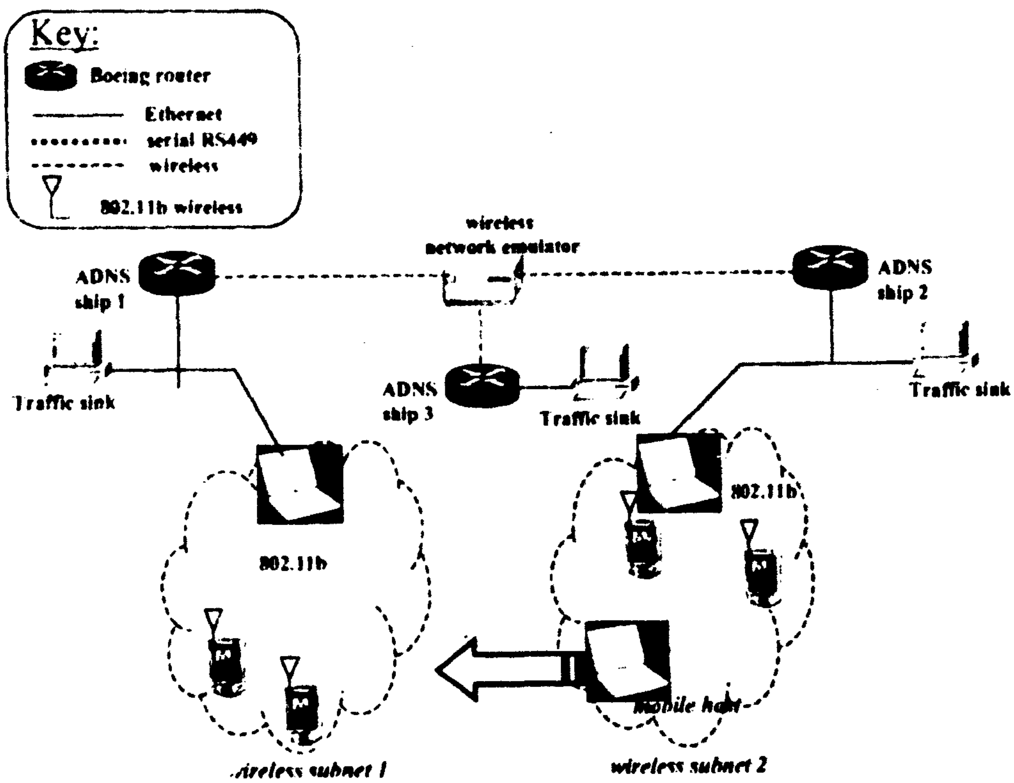


Figure 5-6 Demonstration configuration for "Host Joining and Authentication"

---

[4] R. Moskowitz, P. Nikander, and P. Jokela, "Host Identity Protocol," Internet-Draft: draft-moskowitz-hip-7, available on-line at http://www.ietf.org/internet-drafts/draft-moskowitz-hip-07.txt

7.3.5.3 Demonstration procedures

To set up the demonstration, we will configure devices on wireless subnets 1 and 2, including 802.11b channels, IP addressing, and routing protocol daemons. The wired/wireless router on subnet 1 will be configured to serve as a DHCP server.

The mobile host will first initiate a data connection to a host on ship 1 or 2 using the Host Identity Payload (HIP). This will consist of a HIP protocol handshake, based on public key cryptography, that will establish session keys for the TCP connection to use IPsec encryption. Next, the mobile host will be taken out of range of wireless subnet 1 (either physically out of range or by reconfiguring the 802.11b channel of operation), and into range of subnet 2. The host will first acquire a new IP address in an authenticated manner, using cryptographic techniques for DHCP found in RFC 3118. Upon obtaining new IP configuration for the new wireless subnet, the host will next signal to the corresponding host (using HIP Readdress protocol) that it has changed its IP address. The HIP protocol allows the remote host to authenticate the wireless host and confirm that the host is indeed the same host but at a different address. After this handshake, encrypted communication between the hosts will resume.

7.3.5.4 Metrics and data collection

This demonstration is focused on illustrating the capability of using HIP, in conjunction with wireless OSPF routing, to perform secure host mobility in a tactical environment. The correct operation of HIP will be observed by inspection of data transfer windows on the laptop and fixed host.

8 Summary

Our NBN Composite Routing program has delivered modifications to the standard OSPFv2 routing protocol. After first conducting a simulation trade study of routing protocol alternatives for future NBN networks, we decided that OSPF modifications would speed the technology transition of mobile ad hoc routing protocols to commercial products. We have studied under simulation, specified, and implemented extensions to OSPFv2, and have validated the implementation performance in lab testing and demonstrations. The results of this research program are promising, and we are working with other groups to attempt to standardize such protocol extensions.

8.1 Technology transfer directions

We expect that OSPFv2 extensions would be used by a future Navy if they were made available in commercial products, such as Cisco routers and JTRS radio/routers. The main focus of our ONR KSA FNC Block 2 program is to transition a wireless interface capability to Cisco routers and DSCP-based routing extensions to JTRS radio software. Additionally, standardization of such extensions would facilitate more vendor acceptance. We plan to support standardization efforts under the Block 2 program.

8.2 Future work

We anticipate that the following topics will be the subject of future work in this effort, under ONR KSA FNC Block 2 program:

- quantifying projected performance in future expected operational scenarios, including the handling of similar quantities and types of internal and external OSPF LSAs found in Naval afloat networks.
- identifying key parameters to study for sensitivity analysis.
- specifying how the extensions could be folded into OSPFv3 for IPv6.
- recommending how the extensions fit into the overall QoS framework for the Navy, including use of network management and QoS policy management tools such as Cisco QoS Policy Manager.
- studying extensions to PIM or source specific multicast protocols to enable dynamic policy based routing based on DSCP or other similar mechanisms;
- studying possible extensions to or use of OSPF, IS-IS, or EIGRP to solve scalability concerns relating to the cross-connect of different AORs;
- combining routing protocol extensions with MPLS and OSPF traffic engineering extensions; and
- consideration of interworking issues for more seamless joint operations.

9 Acknowledgments

We thank the following individuals and organizations for assistance in this contract:

- Eric Otte and David Leung of SPAWAR SSC SD, NTII Laboratory, for hosting our May 2003 demonstration, and producing a test report.
- Jerry Ferguson, ONR program manager, for hosting our September 2003 demonstration, again in San Diego.
- Cisco Federal Systems, including Fred Baker, Pepe Garcia, Scott Harris, Steve Kapp, and Dave West, for providing technical assistance and demonstration support during this program, including the loan of Cisco Mobile Access Routers (MAR) for our demonstration, and
- Deborah Goldsmith of MITRE for hosting us at the Navy Quality-of-Service Working Group meeting in June 2003.